



SBCM2021

Simpósio Brasileiro de Computação Musical

Proceedings of the 18th brazilian
symposium on computer music

compmus.org.br/2021

Recife, PE

24 - 27 October, 2021

ISSN 2175-6759 • ISBN 978-65-87003-79-5

GIORDANO CABRAL
FILIPE CALEGARIO
TIAGO FERNANDES TAVARES
IVAN SIMURRA
FLÁVIO LUIZ SCHIAVONI
CRISTIANO FIGUEIRÓ
ISABEL NOGUEIRA
CLAUDIO ROGERIO GOMES DA SILVA
GEBER RAMALHO
JUCIANE ARALDI BELTRAME
ALESSANDRA ALELUIA ALVES
ANDERSON RUFINO DOS SANTOS SILVA
FLAVIANO DIAS FONTES
GUSTAVO TENÓRIO CARNEIRO
HORHANNA ALMEIDA DE OLIVEIRA
JÁDER ANDERSON OLIVEIRA DE ABREU
JOÃO PEDRO MENDES DE OLIVEIRA
JOÃO TRAGTENBERG
MÁRCIA VANDERLEI
MARIANA PEREIRA LELLIS
MYCHELLINE SOUTO CUNHA
NATHÁLIA FERNANDES
RICARDO BRAZILEIRO
TALES HUMBERTO DE AQUINO BORATTO

**PROCEEDINGS OF THE 18TH BRAZILIAN
SYMPOSIUM ON COMPUTER MUSIC**

Dados Internacionais de Catalogação na Publicação (CIP)

S612 Simpósio Brasileiro de Computação Musical (18. : 2021 : Recife, PE)
Anais do 18º Simpósio Brasileiro de Computação Musical [recurso eletrônico] – organizadores: Giordano Cabral ... [et al.] – Porto Alegre : SBC, 2022.

ISBN 978-65-87003-79-5

1. Computação. 2. Música. I. Cabral, Giordano. II. Calegario, Filipe. III. Tavares, Tiago Fernandes. IV. Simurra, Ivan. V. Schiavoni, Flávio Luiz. VI. Figueiró, Cristiano. VII. Nogueira, Isabel. VIII. Silva, Claudio Rogerio Gomes da. IX. Ramalho, Geber. X. Beltrame, Juciane Araldi. XI. Alves, Alessandra Aleluia. XII. Silva, Anderson Rufino dos Santos. XIII. Fontes, Flaviano Dias. XIV. Carneiro, Gustavo Tenório. XV. Oliveira, Horhanna Almeida de. XVI. Abreu, Jáder Anderson Oliveira de. XVII. Oliveira, João Pedro Mendes de. XVIII. Tragtenberg, João. XIX, Vanderlei, Márcia. XX. Lellis, Mariana Pereira. XXI. Cunha, Mychelline Souto. XXII. Fernandes, Nathália. XXIII. Brasileiro, Ricardo. XXIV. Boratto, Tales Humberto de Aquino. XXV. Título.

CDU 004:78

18º Brazilian Symposium on Computer Music

Federal University of Pernambuco
24-27 October 2021
Recife - PE - Brazil

Proceedings

ISBN - 978-65-87003-79-5
ISSN - 2175-6759

Edited by

Sociedade Brasileira de Computação - SBC

Organized by

Giordano Cabral (UFPE/CESAR)
Filipe Calegario (UFPE)
Tiago Fernandes Tavares (UNICAMP)
Ivan Simurra (UFAC)
Flávio Luiz Schiavoni (UFSJ)
Cristiano Figueiró (UFBA)
Isabel Nogueira (UFRGS)
Claudio Rogerio Gomes da Silva (UNIFAP)
Geber Ramalho (UFPE/CESAR)
Juciane Araldi Beltrame (UFPB)
Alessandra Aleluia Alves (UFPE)
Anderson Rufino dos Santos Silva (UFPE)
Flaviano Dias Fontes (UFPE)
Gustavo Tenório Carneiro (UFPE)
Horhanna Almeida de Oliveira (UFPE)
Jáder Anderson Oliveira de Abreu (UFPE)
João Pedro Mendes de Oliveira (UFSJ)
João Tragtenberg (Batebit)
Márcia Vanderlei (UFPE)
Mariana Pereira Lellis (UFSJ)
Mychelline Souto Cunha (UFPE)
Nathália Fernandes (UFPE)
Ricardo Brasileiro (UFPE)
Tales Humberto de Aquino Boratto (UFJF)

Proceedings Editorship

Tales Humberto de Aquino Boratto
Mariana Pereira Lellis
Flávio Luiz Schiavoni

Cover Art

Horhanna Almeida de Oliveira
Mariana Pereira Lellis

Permission to make digital or hard copies of all part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© Brazilian Computer Society (Sociedade Brasileira de Computação – SBC)

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

Sponsors and Organization



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO



Committees

ORGANIZING COMMITTEE

GENERAL CO-CHAIRS

Giordano Cabral (UFPE/CESAR)

Filipe Calegario (UFPE)

TECHNICAL PAPERS CHAIRS

Tiago Fernandes Tavares (UNICAMP)

Filipe Calegario (UFPE)

MUSICAL PAPERS AND FESTIVAL CHAIRS

Ivan Simurra (UFAC)

Flávio Luiz Schiavoni (UFSJ)

Cristiano Figueiró (UFBA)

Isabel Nogueira (UFRGS)

NATIONAL SCHOOL OF COMPUTER MUSIC CHAIR

Cláudio Rogério Gomes da Silva (UNIFAP)

INDUSTRY TRACK CHAIR

Geber Ramalho (UFPE/CESAR)

LOCAL ARRANGEMENTS COMMITTEE

Alessandra Aleluia Alves

Anderson Rufino dos Santos Silva

Flaviano Dias Fontes

Gustavo Tenório Carneiro

Horhanna Almeida de Oliveira

Jáder Anderson Oliveira de Abreu

João Pedro Mendes de Oliveira

João Tragtenberg

Juciane Araldi Beltrame

Márcia Vanderlei

Mariana Pereira Lellis

Mychelline Souto Cunha

Nathália Fernandes

Ricardo Brasileiro

Tales Humberto de Aquino Boratto

Scientific Track Committee

Adriano Monteiro, Universidade Federal de Goiás, Brazil

Alexandre Porres, USP, Brazil

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

Anesio Costa Neto, IFSP - Votuporanga, Brazil

Charles Figueredo de Barros, Universidade Federal de São João del Rei, Brazil

Clayton Mamedes, Federal University of Paraná, Brazil

Cláudio Gomes, Universidade Federal do Amapá, Brazil

Diego Colombo Dias, Universidade Federal de São João del-Rei, Brazil

Edilson Ferneda, Universidade Católica de Brasília - UCB, Brazil
Eduardo Miranda, University of Plymouth, Great Britain
Evandro Miletto, Inst. Fed. do Rio Grande do Sul - Campus Porto Alegre, Brazil
Fabio Kon, IME - USP, Brazil
Filipe Calegario, Universidade Federal de Pernambuco, Brazil
Flávio Schiavoni, Federal University of São João Del Rei, Brazil
Flávio Soares Corrêa da Silva, Universidade de São Paulo, Brazil
Gabriel Rimoldi, Universidade Estadual de Campinas, Brazil
Giordano Cabral, UFPE, Brazil
Isabel Barbancho, University of Malaga, Spain
Isabel Nogueira, Universidade Federal do Rio Grande do Sul, Brazil
Ivan Simurra, UNIVERSIDADE FEDERAL DO ACRE, Brazil
James Correa, Universidade Federal de Pelotas, Brazil
Jean Bresson, Ableton, Germany
Jean-Pierre Briot, LIP6 (CNRS - Univ. Paris 6) & PUC-Rio, France
Joe Timoney, Maynooth University, Ireland
Jose Fornari, State University of Campinas - UNICAMP, Brazil
Juliano Foleiss, Universidade Tecnológica Federal do Paraná, Brazil
Leandro Costalonga, UFES, Brazil
Luis Jure, Universidad de la República, Uruguay
Luiz Naveda, State University of Minas Gerais, Brazil
Manuel Falleiros, UNICAMP, Brazil
Marcelo Pimenta, Universidade Federal do Rio Grande do Sul, Brazil
Marcelo Queiroz, University of São Paulo, Brazil
Marcelo Wanderley, McGill University, Canada
Maria Helena de Lima, UFRGS, Brazil
Martín Rocamora, Universidad de la República, Uruguay
Mikhail Malt, Ircam, France
Rael Toffolo, Estate University of Maringá, Brazil
Rodolfo Coelho de Souza, Universidade de São Paulo, Ribeirão Preto, Brazil
Sanjay Majumder, Pennsylvania State University, USA
Stephan Schaub, Unicamp, Brazil
Tairone Magalhães, Universidade Federal de Minas Gerais, Brazil
Tiago Tavares, Universidade Estadual de Campinas, Brazil

Art Track Committee

Adriano Monteiro, Universidade Federal de Goiás, Brazil
Ariane Stolfi, Universidade de São Paulo, Brazil
Cristiano Figueiró, <https://ihac.ufba.br/>, Brazil
Danilo Rossetti, Universidade Federal de Mato Grosso, Brazil
Fernando Iazzetta, Universidade de São Paulo - USP, Brazil
Flávio Schiavoni, Federal University of São João Del Rei, Brazil
Igor Maia, Universidade Federal de Minas Gerais, Brazil
Isabel Nogueira, Universidade Federal do Rio Grande do Sul, Brazil
Ivan Simurra, UNIVERSIDADE FEDERAL DO ACRE, Brazil
James Correa, Universidade Federal de Pelotas, Brazil
Ricardo Bordini, UFBA, Brazil
Ricardo Dal Farra, Concordia University, Canada

Comitê da Escola Nacional de Computação Musical

Aluizio Oliveira, Universidade Federal de Minas Gerais, Brazil
Charles Figueredo de Barros, Universidade Federal de São João del Rei, Brazil
Cláudio Gomes, Universidade Federal do Amapá, Brazil
David Borgo, University of California San Diego (UCSD), USA
Edilson Ferneda, Universidade Católica de Brasília - UCB, Brazil
Fabio Kon, IME - USP, Brazil
Fernando Cruz, Universidade de Brasília, Brazil
Filipe Calegario, Universidade Federal de Pernambuco, Brazil
Flávio Schiavoni, Federal University of São João Del Rei, Brazil
Flávio Soares Corrêa da Silva, Universidade de São Paulo, Brazil
Hugo de Paula, PUC Minas, Brazil
Ivan Simurra, UNIVERSIDADE FEDERAL DO ACRE, Brazil
Jean-Pierre Briot, LIP6 (CNRS - Univ. Paris 6) & PUC-Rio, France
Manuel Falleiros, UNICAMP, Brazil
Marcelo Pimenta, Universidade Federal do Rio Grande do Sul, Brazil
Marcelo Wanderley, McGill University, Canada
Marcos Laia, Universidade Federal de São João Del Rei, Brazil
Maria Helena de Lima, UFRGS, Brazil
Martín Rocamora, Universidad de la República, Uruguay
Monica Vermes, Universidade Federal do Espírito Santo, Brazil
Rael Toffolo, Estate University of Maringá, Brazil
Ricardo Dal Farra, Concordia University, Canada
Tiago Tavares, Universidade Estadual de Campinas, Brazil

Prefácio

O XVIII Simpósio Brasileiro de Computação Musical (SBCM) ocorreu em Recife de 24 a 27 de Outubro de 2021. Quando dizemos que ocorreu em Recife é modo de dizer pois, na verdade, em 2021, o SBCM aconteceu em diversos lares do Brasil e do exterior, pois a pandemia da COVID 19 não nos permitiu realizar o evento de maneira presencial. A organização do evento contou com a coordenação dos professores Giordano Cabral e Filipe Calegario, de uma equipe de colaboradores e colaboradoras da UFPE, além de todo o apoio e logística do CIn (Centro de Informática) da UFPE e do CESAR (Centro de Estudos e Sistemas Avançados do Recife). A organização remota permitiu unir Rio Grande do Sul, Acre, Bahia, Amapá, Minas Gerais, São Paulo e Paraíba e contou com a coordenação de Tiago Tavares (UNICAMP), Ivan Simurra (UFAC), Flávio Schiavoni (UFSJ), Cristiano Figueiró (UFBA), Isabel Nogueira (UFRGS), Cláudio Rogério Gomes da Silva (UNIFAP), Juciane Araldi Beltrame (UFPB) e Geber Ramalho (UFPE/CESAR), além de uma grande equipe que prestou um enorme serviço para que tudo estivesse funcionando no evento.

Se o formato remoto impediu o encontro presencial, tão importante para eventos como este, ele permitiu que o evento ganhasse outras proporções. Pensando nisso, a equipe organizadora colocou como meta a mobilização da comunidade de computação musical centrada no Brasil, o aumento da relevância e da visibilidade da pesquisa feita no Brasil, a inclusão de novos perfis e públicos e de diálogo com áreas afins. Assim, baseado nestas diretrizes, foram criadas três trilhas, que complementaram a **trilha científica** (tanto em computação quanto em música), já tradicionais.

Para atender a necessidade de incluir novos pesquisadores e novas pesquisadoras, em especial, discentes de graduação em computação e artes, o SBCM sediou a primeira **Escola Nacional de Computação Musical**. Este evento foi um pontapé inicial para a criação de escolas regionais e de novas escolas nacionais, a ocorrer seja simultaneamente seja alternadamente ao simpósio. A escola proporcionou a submissão e apresentação de trabalhos de pesquisa em andamento, a realização de cursos e workshops, demonstração de projetos e a apresentação de reports de grupo de pesquisa para permitir uma visão geral da pesquisa de computação e música no Brasil. Os **artigos curtos** foram classificados em macro sessões de: educação (dois artigos), linguagens computacionais e musicais (dois artigos), sistemas interativos

(cinco artigos) e recuperação de informação musical (cinco artigos). Sobre os **workshops**, apresentaram diversos objetivos desde tutoriais de iniciação à computação musical, criatividade visual e musical e ainda com troca de ideias futuras entre pesquisadores.

Para aumentar o diálogo com as artes, foi institucionalizado o **Festival da SBCM**, com apresentações artísticas, tanto de artistas convidados e convidadas quanto inscritos e inscritas no evento. As atividades artísticas já tinham uma tradição no SBCM em suas edições anteriores, mas com a XVIII edição foi institucionalizada sob o nome de Festival. O Festival estimulou a diversidade de artistas proponentes incluindo e incentivando a participação de artistas mulheres, artistas periféricos/periféricas, negros/negras, da comunidade LGBTQIA+ e pessoas não binaries. O evento incluiu composição musical e artes digitais relacionadas à música em sua prática no sentido mais amplo do termo (instalações, performances, webart, sistemas artísticos interativos, som e música nas artes visuais, telemática, entre outros).

Tentando ampliar ainda mais as possibilidades de interseções da computação musical e sua comunidade, foi criada uma **trilha da indústria**. Uma aproximação com a indústria tem se mostrado essencial, tanto para buscar investimentos, quanto para aumentar a relevância da pesquisa e para ter acesso a avanços tecnológicos realizados em empresas. Nesta edição, foi feita, uma mesa redonda temática, a disponibilidade de desafios propostos por empresas, e uma job-fair, na qual empresas apresentaram seus objetivos, perfis de profissionais que lhes interessam, vagas de trabalho e, ao final, conversas com possíveis interessados em trabalhar com eles.

Por fim, foi proposto um desafio Wikipedia durante o evento onde os participantes e as participantes do SBCM foram convidados e convidadas a colaborar com a Enciclopédia aberta e auxiliar na difusão e divulgação de conhecimentos em português sobre o tema. Obteve-se a cerca de 3500 incrementos e/ou modificações no Wikipedia contabilizadas a partir do lançamento do desafio, um mês antes do evento.

A tentativa de ampliar a participação da comunidade se mostrou bastante frutífera e o evento contou com 102 inscritos e inscritas e cerca de 95 participantes sendo um participante da Finlândia, dois da Turquia, um de Portugal e dois dos Estados Unidos. A trilha científica contou com 25 artigos científicos, quatro artigos artísticos e 14 artigos curtos aprovados, com média de três autores ou autoras por artigo, com

participação feminina em cinco artigos. O evento contou ainda com quatro palestrantes, entre convidados e convidadas da computação e da música.

A ENCM contabilizou cinco relatórios de grupos de pesquisas dos estados do Amapá, Espírito Santo, Minas Gerais e São Paulo. Esses relatórios destacaram ações realizadas desde sua origem e, principalmente, durante o período de 2020 a 2021, em momento pandêmico. Notou-se grande variedade entre os grupos apresentados, desde grupos em formação inicial, em fase de consolidação e por grupos consolidados por décadas. A ENCM contou também com duas demonstrações de projetos com características individuais com aspectos de análise e filosofias de diferentes aspectos computacionais, visuais e musicais além de oito workshops, com carga horária total de 37h, obtendo-se o total de 195 participantes com média de 15% de participação feminina. Estes workshops foram realizados de forma concomitante e em maioria com gratuidade.

No contexto do Festival, a edição do SBCM 2021 organizou duas **Mesas Redondas** considerando temas acerca das práticas musicais em esferas amplas e inclusivas dentro do universo musical criativo e tecnológico que estão presentes, atualmente, em todos os ambientes de diálogo artístico e musical. O evento teve ainda oito performances artísticas convidadas e nove apresentações de participantes do evento.

O evento ainda fez questão de premiar como **Best lab report**: “A prospective report on the research developed at the Laboratory of Audio and Music Technology at USP” de Regis Faria (University of São Paulo), Ricardo Thomasi (ECA/USP), João Monnazzi (University of São Paulo), Eduardo Bonachela (EACH/USP), Andre Giolito (EACH/USP) e Gabriel Lemos (ECA/USP); **Best short paper**: “Investigando Tecnologias Educacionais para o Ensino e Aprendizagem de Música para Deficientes Visuais Totais” de Sara Remedios (UFRJ), Sandro Luis Freire de Castro Silva (UFRJ) e Simone Bacellar Leal Ferreira (UFRJ); e Best Art paper: “Playing Time-Variant Audio Feedback Networks” de Adriano Claro Monteiro (Universidade Federal de Goiás). Foi ainda premiado o artigo “Tempo estimation via neural networks a comparative analysis” de Mila Soares de Oliveira de Souza (UFRJ), Pedro Nuno de Souza Moura (UFRJ) e Jean-Pierre Briot (LIP6 CNRS - Univ. Paris 6) como **Best Women in Science Paper**. O evento premiou ainda Rômulo Vieira (UFSJ), como **Best Wikipedia Contributor**, e Tiago Tavares (UNICAMP) como melhor membro da organização.

Esta edição do SBCM ainda vai dar muito trabalho para os participantes pois será publicado uma edição especial da **Revista Vórtex** com artigos musicais selecionados no evento, uma edição especial da **Revista Eletrônica de Iniciação Científica** (REIC) com artigos selecionados da ENCM, uma edição especial da **Journal of Communication and Information Systems** (JCIS) e uma edição especial da **Revista IEEE Latin America** com artigos selecionados da Trilha científica do evento.

Ao fim do evento, ficou a certeza de que o mesmo foi uma lufada de ar fresco em nossos pulmões já tão cansados de pandemia e isolamento social. Muitos e muitas participantes que tiveram a oportunidade de estar em seu primeiro SBCM relataram que o evento foi sensacional. Nós da organização só podemos agradecer a todos, todes e todas que confiaram, se inscreveram, participaram e contribuíram para que o evento acontecesse da melhor maneira possível. Muito obrigado!

Organização SBCM 2021

Invited Talks

DEEP LEARNING FOR MUSIC GENERATION: ORIGINS, SUCCESSES AND CHALLENGES

Sunday, October 27, 18:00 – 19:30

SPEAKER: JEAN-PIERRE BRIOT

<https://youtu.be/wQjpSA32Ys4>



A growing area of application of the current wave of deep learning (the hyper-vitamed return of artificial neural networks) is the generation of creative content, notably the case of music (and also images and text). The motivation is in using machine learning techniques to automatically learn musical styles from arbitrary musical corpora and then to generate musical samples from the estimated distribution, with some degree of control over the generation. In this talk, we will at first analyze some early works from the late 1980s using artificial neural networks for music generation and how

their pioneering contributions have prefigured recent techniques (e.g., hierarchical models and Deep Dream). We will then present some recent achievements using last generation architectures such as VAE, GAN and Transformer and analyze successes and challenges. Last, but not the least, will also address the issue of the authorship of the generated music and corresponding rights issues. Jean-Pierre Briot is a senior researcher in computer science at CNRS (Centre National de la Recherche Scientifique) and Sorbonne Université in Paris, France. He is also permanent visiting professor at PUC-Rio and has recently been visiting professor at UNIRIO, both in Rio de Janeiro, Brazil. His general research interests are about the design of intelligent adaptive and cooperative software, at the crossing of artificial intelligence, distributed systems and software engineering, with various application fields such as Internet of Things, decision support systems and computer music. He is the first author of a recent reference book on the use of deep learning (artificial

intelligence/machine learning) techniques for music generation. His current interest is focused on AI and music creativity. Jean-Pierre Briot holds a masters in mathematics (1980), a doctorship (PhD) in computer science (1984) and an “habilitation à diriger des recherches” in computer science (1989), all from Université Pierre et Marie Curie (aka Paris VI, since 2018 renamed/merged as Sorbonne Université). He also holds degrees in music, music acoustics and Japanese language. He has been visiting researcher in various institutions (Kyoto University, Tokyo Institute of Technology, University of Illinois at Urbana-Champaign, University of Southern California, University of Tokyo...). He has advised more than 25 PhD students. He has edited 12 books or journal special issues. In 2010, he has created the CNRS permanent representation office in Rio de Janeiro, for scientific cooperation with Southern America. For more details (including access to publications), please see <http://www-desir.lip6.fr/~briot/cv/>

MUSIC SOURCE SEPARATION: CHALLENGES AND SOLUTIONS **SEPARAÇÃO DE FONTES MUSICAIS: APLICAÇÕES E DESAFIOS**

Monday, October 25, 18:00 – 19:30

SPEAKER: IGOR PEREIRA

<https://youtu.be/Yx9jFE4TI0Q>



Igor Pereira é head of Machine Learning no aplicativo Moises, onde lidera o time que desenvolve algoritmos inteligentes para recuperação de informação musical – com enfoque na separação de fontes musicais. Igor é Doutor em Engenharia Elétrica e Computação pela Universidade Federal do Rio Grande do Norte (2019), com estágio de Doutorado Sanduíche (SWE) no Institute of Applied Sciences and Intelligent Systems (ISASI) na Itália (2018). Em seu doutorado, pesquisou sobre estratégias de sincronização de sistemas multimídia com a utilização de algoritmos de Machine Learning, desenvolvendo

métodos de estado da arte que foram publicados em periódicos internacionais. Também é Mestre em Engenharia Elétrica pela Universidade Federal da Paraíba (2014) e graduado em Engenharia Elétrica pelo Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (2011). Igor tem experiência em: Recuperação de Informação Musical, Sistemas Distribuídos, Redes Neurais Artificiais, Aprendizagem de Máquina, ip-cores, FPGAs e Sistemas Embarcados.

ASPECTOS HUMANOS DA ECONOMIA PSÍQUICA DOS ALGORITMOS

Tuesday, October 26, 18:00 – 19:30

SPEAKER: DANI RIBAS

Fotografia: Patrícia Soransso

<https://youtu.be/IdPRrk4JlGw>



Dani Ribas é diretora da Sonar Cultural Consultoria. É Doutora em Sociologia pela UNICAMP, e a partir de sua tese criou o método ID_Musique, que analisa e incorpora o comportamento do público às estratégias criativas e de negócios de artistas da música. Foi consultora da UNESCO e do Mercosul Cultural, foi pesquisadora do CPF-SESC SP, foi diretora do DATA SIM e fez pesquisa em Economia Criativa para o IPEA. Integra a Rede SateliteLAT de Mulheres na Indústria da Música Latino-americana. É professora de Music Business no Music Rio Academy, OnStage Lab, FESP SP, Música & Negócios PUC-Rio e de Gestão Cultural na UNICAMP. É consultora em

planejamento e gestão de carreira na música, com base em análise de dados e tendências de comportamento de público.

DESIGNING NIMES FOR WIDESPREAD USE

Wednesday, October 27, 18:00 – 19:30

SPEAKER: MARCELO M. WANDERLEY

<https://youtu.be/19C2Uw6z19M>



New interfaces for musical expression (NIME) allow for unprecedented access to sound parameters and the possibility to interact with sound and music on a variety of levels: from acoustic instrument simulations to interfaces for conducting pre-recorded music (e.g. interactive sequencers) to laptop orchestras/live coding. Despite hundreds of examples of interfaces proposed in the last decades, very few of these have been played by more than a dozen individuals. This talk will discuss challenges and opportunities involved in the design of NIMEs

aiming at a large base of users. Could NIMEs become widespread tools for musical expression or are musical interfaces doomed by the drive for novelty (the “N” in NIME)? To provide answers to this question, I will discuss several examples of interfaces developed at the Input Devices and Music Interaction Laboratory (IDMIL), ranging from one-off prototypes to DMIs produced in small numbers (dozens of copies), focusing on issues inherently related to academic institutions’ structures, the availability of resources, and the need for specialized expertise to produce commercial-level instruments. Marcelo M. Wanderley is Professor of Music Technology at McGill University, Canada and International Research Chair at Inria Lille – Nord Europe, France. His research interests include the design and evaluation of digital musical instruments and the analysis of performer movements. He co-edited the electronic book “Trends in Gestural Control of Music”, 2000, co-authored the textbook “New Digital Musical Instruments: Control and Interaction Beyond the Keyboard”, 2006, and chaired the 2003 International Conference on New Interfaces for Musical Expression (NIME03). He is a member of Computer Music Journal’s Editorial Advisory Board and a senior member of the ACM and of the IEEE.

The Festival

Sunday / Domingo- 19:30

OTOCANTO

(MARINA MAPURUNGA)

Artista e pesquisadora que atua no campo do audiovisual, da arte sonora e da música. Professora de som dos cursos de Cinema e Artes Visuais da Universidade Federal do Recôncavo da Bahia (UFRB). Doutoranda em Música (Sonologia) pela USP, onde desenvolve sua pesquisa sobre práticas sonoras experimentais como estratégias de ensino para a criação sonora em cursos de cinema e audiovisual. Pesquisadora do NuSom - Núcleo de Pesquisas em Sonologia da USP e do LinkLivre - Grupo de Estudos e Práticas Laboratoriais em Plataformas, Softwares Livres e Multimeios da UFRB. Integrante da rede Sonora - músicas e feminismos e da Orquestra Errante. Coordenadora e integrante do Laboratório de Pesquisa, Prática e Experimentação Sonora - SONatório, projeto de extensão da UFRB, e da OLapSo (Orquestra de Laptops SONatório - UFRB). Website: www.mapu.art.br. email: marinanimula@gmail.com

BETAMAXERS

(ISABEL NOGUEIRA e LUCIANO ZANATTA)

Betamaxers é um duo formado por Isabel Nogueira e Luciano Zanatta. Ambos são produtores, músicos, sound designers e videomakers. Suas obras em áudio e vídeo integram improvisação, linguagens experimentais e algorítmicas e meios eletrônicos. Os procedimentos incluem programação, sampleamento e síntese de som e imagem em performances ao vivo ou sequenciadas.

Betamaxers is a duo formed by the musicians Isabel Nogueira e Luciano Zanatta. Both are producers, musicians, sound designers and videomakers. The duo audio and video works integrate improvisation, experimental and algorithmic languages and electronic media. Procedures include programming, sampling and sound and image synthesis in live or sequenced performances.

THE NOISE INSIDE US

(MICAEL ANTUNES DA SILVA, MARIANA CABRAL)

Micael Antunes is a musician and composer who graduated from FAAM (Alcantara Machado Arts School) and a Master's degree at the University of Sao Paulo. Nowadays, he is a doctoral student at the Arts Institute of the University of Campinas under the supervision of Jonas Manzolli and Danilo Rossetti, with collaborative project research with the Interdisciplinary Nucleus for Sound Communication (NICS-Unicamp). He worked as a teacher at the Guri Santa Marcelina project, taught theory, improvisation courses, and giving teacher training. His academic research includes tuning systems, psychoacoustics models, and musical analysis, currently applied to his compositions. He participated as a composer and music producer in the compositions that were released with Pedro Marques' book "Cena Absurdo", published in 2016. His music was performed at events such as the International Percussion Festival at EMESP (2016) and "Musica Estranha Festival" in 2014 and 2015, in Sao Paulo, III Congreso Internacional de Ciencia y Tecnologia Musical, in Buenos Aires. He participated as an author of academic papers in events such as the 15th Brazilian Symposium on Computer Music, in Campinas, and the 13th International Symposium on Computer Music Multidisciplinary Research, in Porto, Portugal. micael.antunes@nics.unicamp.br

Mariana Cabral is a percussionist and sound artist, studying at the Federal University of Mato Grosso do Sul. She played at the Percussion and Batucaria Laboratory (LAPEBA-UFMS). She was a percussionist for the Choir of Trombones and the Symphonic Band at the Federal University of Mato Grosso do Sul. She develops her artistic research focused on sound art and interaction with electronic devices to support the creation of artistic installations. cabralmariana030@gmail.com

A TELEMATIC PERFORMANCE WITH WORC

(ALI SERKAN SEVILGEN)

<http://sbcm2021.serkansevilgen.com/>

Serkan Sevilgen is an Istanbul-based computer programmer and composer. He holds a MA degree in Sonic Arts from Istanbul Technical University Center for Advanced Studies in Music. He is a member of the Istanbul Coding Ensemble (ICE) It was founded in 2018 with a focus on improvisation with musical

algorithms using ?just-in-time? programming techniques and real-time communication with ad-hoc network music systems. His interests are computer music, algorithmic composition, sonification, improvisation, network music, and spatial audio.

SYMPHONY IN BLUE 2.0

(KONSTANTINOS VASILAKOS, ALI SERKAN SEVILGEN, JERFI AJI, SCOTT WILSON)

Konstantinos Vasilakos is an interdisciplinary artist working in the milieu of digital art and interactive media, and Electroacoustic Music. He is based in Istanbul and leads the Sonic Arts postgraduate studies path at the Center for Advanced Studies in Music MIAM, in Istanbul Technical University. His courses focus on the creation of interactive music, sound installations and algorithmic/non-linear composition. His research interests include sonification, new instruments for musical expression and live-coding with networked music systems.

Serkan Sevilgen is an Istanbul-based computer programmer and composer. He holds a MA degree in Sonic Arts from Istanbul Technical University Center for Advanced Studies in Music. He is a member of the Istanbul Coding Ensemble (ICE) It was founded in 2018 with a focus on improvisation with musical algorithms using ?just-in-time? programming techniques and real-time communication with ad-hoc network music systems. His interests are computer music, algorithmic composition, sonification, improvisation, network music, and spatial audio.

TIME GARDEN

(CHARLES NICHOLS, ZACH DUER, SCOTTY HARDWIG)

Zach Duer is an Assistant Professor teaching in the Creative Technologies Program in the School of Visual Arts at Virginia Tech in Blacksburg, Virginia. His work lies at a series of intersections: sound and visualization; careful composition and improvised performance; intuitive musical spontaneity and structured digital systems. Scotty Hardwig is a movement and media artist whose work investigates the spaces between the human and the technological, the real and the digital, the body and the environment, and the anatomical/evolutionary and the social. He is an active creator of contemporary works for stage and screen, and an educator teaching courses in movement,

performance and integrated media at Virginia Tech. Composer, violinist, and computer music researcher Charles Nichols (www.charlesnichols.com) explores the expressive potential of instrumental ensembles, computer music systems, and combinations of the two, for the concert stage, and collaborations with dance, video, and installation art. His research includes spatial audio, data sonification, motion capture for musical performance, and telematic performance. He teaches Composition and Creative Technologies at Virginia Tech and is a Faculty Fellow of the Institute for Creativity Arts and Technology.

Monday / Segunda – 21:00

CODE TO CAGE - FROM THE ANARCHY OF SILENCE, WE DEAL WITH THE CAUSAL NOISE.

(ANDREA MAY)

<https://codetocage.blogspot.com>

Andrea May (BRASIL) é artista visual e sonora, curadora independente e mestra em Artes Visuais pela Universidade Federal da Bahia (UFBA). Atualmente é professora substituta no curso de Artes Visuais do Centro de Artes, Humanidades e Letras/ Universidade Federal do Recôncavo da Bahia (UFRB). Também conhecida como May HD, seu projeto solo “Noisy Turntablism” (Discotecagem ruidosa) consiste na livre improvisação utilizando discos de vinil preparados ou destruídos que, somados a efeitos eletrônicos conectam o experimentalismo à potência do sensível feminino em criações sonoras viscerais. Com esta proposta já se apresentou no Festival NOVAS FREQUÊNCIAS Ano X, Festival DYSTOPIE 2020 (Berlin), Circuito de Música Contemporânea – CMC (2018 e 2019), Festival DIGITÁLIA, CURTO CIRCUITO na Audio Rebel (RJ); Centro da Terra (SP), Fauhaus (SP), Casa Híbrido (BH), desenvolvendo parcerias com artistas como Unix 11, Ida Toninato (Canadá), Bella, Marcela Lucatelli, Nahnati Francischini, Aishá Roriz, Edbrass Brasil, Coletivo MACCHINA SOM ALLSTARS, dentre outros. Álbuns lançados: AUTOSAVE (Noise Invade); SONHO DOBRADO (Pan Y Rosas Discos – Chicago/ EUA), ORBITAR (single) e EXPULSO (Al revés). <http://www.andreamay.com.br/p/about.html>.

ORDER AND PROGRESS: A SONIC SEGUE ACROSS A AURIVERDE

(ANGELO FRAIETTA)

bio: Angelo started playing the guitar at the age of 15 while completing an Electronics Technician apprenticeship in the Royal Australian Air Force. Since that time, music and electronics have been an integral part of Angelo's creative being. Angelo has been designing and building electronic musical instruments and software for use within the arts community since 2000, providing engineering and artistic support for artists and universities on an international scale. Many artists and universities have used his electronics and services in exhibitions and in various venues. He is often approached to create technology solutions from industries ranging from medical, legal, academic, industrial, transcription and the arts. Angelo is an amateur astronomer and often looks for ways of combining science with art.

SOBRE A ELETRODINÂMICA DOS CORPOS DE TRABALHO

(JEFF MORRIS)

bio: Jeff Morris creates musical experiences that engage audiences minds with their surroundings. His performances, installations, lectures, and writings appear in international venues known for cutting-edge arts and deep questions in the arts. He has won awards for making art emerge from unusual situations: music tailored to architecture and cityscapes, performance art for the radio, and serious concert music for toy piano, robot, Sudoku puzzles, and paranormal electronic voice phenomena. He has presented work in the Onassis Cultural Center (Athens), Triennale Museum (Milan), D-22 (Beijings avant-garde music scene), the International Symposium on Electronic Art the Network Music Festival, the Lyndon B. Johnson Presidential Library and Museum (Austin), the Chicago Architecture Foundations Open House Chicago, the Boston Microtonal Society, and the Network Music Festival. His work has won awards in the Concours de Bourges (France), Viseu Rural (Portugal), Music in Architecture International Competition (Austin), the UnCaged Toy Piano Competition (NYC), and the Radio Killed the Video Star Competition (NYC). His music is on Ravello Records.

HOMINO-IDEA

(ANESIO COSTA NETO, AUGUSTO AMSTALDEN, TIAGO TAVARES)

<https://motus.art.br/homino-idea>

Anésio Neto (a.k.a. stellatum_) is a Philosophy teacher at Federal Institute of São Paulo [Instituto Federal de São Paulo (IFSP)] and Ph.D. in Sound and Visual arts for Universidade de Brasília (UnB), Brasília, Brazil. Under the name “stellatum_,” Anésio Neto explores sounds, either recorded from Cerrado (ecosystem from Brazil) or produced by himself. By combining different sonic characteristics into ambiances, stellatum_ aims to expand one’s perception of Nature by offering cues to perceive its own length of time. stellatum_ explores sonic perception to create an expanded ambient wherein people could experience an amount of Nature’s complexity that underlies what is possible to perceive. stellatum_’s primary intention as an art researcher is to enhance the sensation of immersiveness toward his sound compositions.

Augusto Almstaden is both a Computer Engineering and Sound Engineering student at University of Campinas (UNICAMP). Although his work in audiovisual interactions is fairly new, he is determined to learn new skills and improve existing ones to maximize his knowledge about this work area. Fascinated by music for a long time, he has found a friendly face in applying his programming skills in a vast range of musically-related software developments.

Tiago Tavares teaches Computer Engineering at the School of Electrical and Computer Engineering (FEEC), University of Campinas. He has worked in audiovisual interactions since 2015, and has a great interest in new technological support for artistic expressions. He has developed camera-based, audio-based, AI-based, and sensor-based interactive devices, and has used them in audiovisual installations and performances. He uses computers, microcontrollers, and electronics sensors and actuators. Tiago is especially interested in making technological support for art broadly accessible, both economically and technically

MISKETS AND CANICAS

(MATIAS VILAPLANA AND ANIL CAMCI)

Matias Vilaplana’s creative practices involve recording/mixing, music composition, musical interaction design in virtual reality and ensemble improvisation with live audio processing. His current research is focused in the

exploration of musical interaction design in virtual environments using consumer grade HMDs and Motion Capture technology employing practice-based methods. He holds a Bachelor of Fine Arts with distinction in Music Technology from Universidad de Chile in Santiago, Chile, as well as an M.A. in Media Arts from University of Michigan. He is currently a PhD student in the Composition and Computer Technologies program at University of Virginia. [<http://matvilap.github.io>] Anıl Çamcı is an Assistant Professor of Performing Arts Technology at the University of Michigan. His work investigates new tools and theories for multimodal worldmaking at the intersection of extended reality, human-computer interaction, spatial audio and electronic music. Previously, he worked at the University of Illinois at Chicago, where he led research projects on interaction design and immersive audio in virtual reality contexts, and Istanbul Technical University, where he founded the Sonic Arts Program. He completed his PhD at Leiden University in affiliation with the Institute of Sonology in The Hague, and the Industrial Design Engineering Department at Delft University of Technology. Çamcı's research and artistic work has been featured in leading journals and conferences. He has been granted several awards, including the Audio Engineering Society Fellowship, ACM CHI Artist Grant, and NIME Best Installation Prize. [<http://anilcamci.com>]

FAREWELL

(MEI-LING LEE)

Taiwanese-born composer Dr. Mei-Ling Lee's work integrates contemporary western music with Asian culture. Her work regularly draws inspirations from western and Chinese poetry. She received her Ph.D. degree in Composition, studied under Dr. Robert Kyr, and Dr. David Crumb. She is currently pursuing her second Doctor of Musical Arts degree in Music performance, emphasis in Performance of Data-driven Instruments, studying under Dr. Jeffrey Stolet. Her work has been performed in various conferences, including ICMC (International Computer Music Conference), ISEA (International Symposium on Electronic Art), SEAMUS (Society for Electro-Acoustic Music in the United States), and KISS (KYMA International Sound Symposium).

Tuesday / Terça - 21:00

“SECRETS, DREAMS, FAITH AND WONDER - A VISUAL-MUSIC FILM?”

(STEPHEN POPE)

Stephen Travis Pope is an award-winning composer, film-maker and computer scientist based in Santa Barbara, California. He has worked developing multimedia software since the early 1980s and has over 100 publications on music theory and composition, computer music and AI. His compositions are available from Perspectives of New Music, CDCM/Centaur Records, The Electronic Music Foundation, Touch Records and HeavenEverywhere Media. He has taught and conducted music research at Stanford University, the University of California Berkeley and Santa Barbara, in Berlin, Stockholm, Amsterdam, Toronto and elsewhere. Stephen is also a practicing Quaker and active clergy in state and federal prisons. stephen@heaveneverywhere.com

AI PHANTASY (2020) ELECTROACOUSTIC SOUND COMPOSITION FOR STEREO OR MULTICHANNEL SPEAKER ARRAY.

(PANAYIOTIS KOKORAS)

Kokoras is an internationally award-winning composer and computer music innovator, and currently an Associate Professor of composition and CEMI director (Center for Experimental Music and Intermedia) at the University of North Texas. Born in Greece, he studied classical guitar and composition in Athens, Greece and York, England; he taught for many years at Aristotle University in Thessaloniki. Kokoras's sound compositions use sound as the only structural unit. His concept of “holophonic musical texture” describes his goal that each independent sound (phonos), contributes equally into the synthesis of the total (holos). In both instrumental and electroacoustic writing, his music calls upon a “virtuosity of sound,” a hyper-idiomatic writing which emphasizes on the precise production of variable sound possibilities and the correct distinction between one timbre and another to convey the musical ideas and structure of the piece. His compositional output is also informed by musical research in Music Information Retrieval compositional strategies, Extended techniques, Tactile sound, Hyperidiomaticity, Robotics, Sound and Consciousness. More information at <http://www.panayiotiskokoras.com>

GITAN

(LÊV1ÄT4 E FLAVIA GOA)

LÊV1ÄT4 - Graduando de design gráfico, artista independente e pesquisador pelo CNPQ em Arte Eletrônica e Programação para o Design e a Arte. Flavia Goa - Guitarrista de improvisação, artista sonora e compositora musical de música eletrônica. Vem trabalhando com trilha sonora para dança desde 2018. Atua em ação social em projeto solidário.

APPROACHES

(KARL GERBER)

He began playing the electric bass autodidactically. He attended musicology lectures as a guest student in 1975 with Riethmüller in Freiburg. Karl F. Gerber ventured his first experiments with music electronics during his apprenticeship as a physics lab technician in the microelectronics industry. After turning to jazz, he studied double bass with Adelhard Roidinger in Munich. He obtained a M.Sc. in physics from the LMU Munich. As a composer he is self-taught, at the same time he considers his courses with H. W. Erdmann, Cort Lippe, Robert Rowe, Carola Bauckholt or piano lessons with Götz Tangerding and Alex Grünwald together with harmony lessons with Joe Haider and Joe Viera as very useful and with gratitude. A graduate physicist, he worked in the microelectronics industry for many years and as a technical school lecturer in Munich until 2017. His interests range from jazz and electronics to interactive computer sound installations: e.g. Perseus, 1987, at Galerie X, Munich. Experiments with real-time mathematical composition began in 1984 on the Commodore C64 and eventually led to live algorithmic performances such as a co-improvisation with the University of Michigan Dancers at ICMC 1998 in Ann Arbor, Michigan, USA. "Beautiful Numbers" received the 2nd prize as electronic "Music for Dance" in Bourges; with release on UNESCO CD "Cultures Electroniques Vol 12". In 2019 this track was selected by two choreographers at once for different dance performances for "New Music in Motion" at Schwere Reiter, Munich. "Stream" was selected by the German Society for Electroacoustic Music for CD7 on Cybele. Since "Loops" for piano solo, he has also been creating works in traditional notation without electronics, such as "VC3e" for harpsichord four-hands. A Siemens/MGNM commission resulted in "Unarieunbegleitet": the voice of the singer influences an acoustic computer

grand piano in a complex way. Here the abandonment of electronic sound synthesis is already apparent – without giving up interactivity! The “experimental violin automaton” was first presented as a work-in-progress in 2012 at the Deutsches Museum in Munich. In the process, Gerber holistically designs both software, hardware, mechatronics and composition itself. Since an invitation to the Kontakte Festival 2017 at the AdK Berlin, his installation “Computer Music without Loudspeakers” has also attracted international interest. So in USA (Boston Berklee College) and South Korea, Seoul 2019. From a large number of applicants his installation “Violinautomat” was selected by the ISCM to the World Music Days in Tallinn, Estonia. Most recently he received the “Award of Distinction” at Matera Intermedia 2020 in Matera Italy in the Performance/Sound Art category.

SUBSTÂNICAS MOLDÁVEIS

(DANILO ROSSETTI)

Daniilo Rossetti is a composer and researcher that focuses his work in the use of technology and interdisciplinary research in creative processes and musical analyses performances. He is author of musical works for different formations (solo or ensembles), acousmatic, live electronics and multi-modal (audiovisual installations, music and dance, networked and telematic music). He is assistant professor at the Department of Arts of the Federal University of Mato Grosso (UFMT), and collaborator professor at the graduate music studies of the Institute of Arts at UNICAMP. He accomplished a post-doc research at the Interdisciplinary Nucleus for Sound Communication, at the University of Campinas (funded by the São Paulo Research Foundation) and earned a Ph.D. in Music Composition at the same university, with a doctoral stage at the CICM/Paris 8 University. His compositions have been played in many events and festivals such as ICMC, CMMR, NYCEMF, NIME, CICTeM, NowNet Arts, BIMESP, SBCM, FILE Hipersônica, ANPPOM, and Funarte Contemporary Music Biennial. He is author and coauthor of several articles concerning creative processes in music and musical analyses.

Wednesday / Quarta - 21:00

MUITA ONDA PRA VINICIUS DE MORAES

(ZÉ BALBINO)

Zé Balbino aka ZEBB é DJ, produtor musical e artista multimídia, e além de produzir dentro de linguagens como vídeo e fotografia, produz arranjos eletrônicos desde 2005. Começou produzindo para experiências radiofônicas, mas, com o nascimento do seu projeto Cidadão Comum Sistema de Som, adentrou o universo da Produção Musical dedicando-se inicialmente ao Reggae, como estilo musical favorito e ao Dub como linguagem, produzindo faixas e projetos musicais com cantoras como a nigeriana Okwei Odili e a brasileira Tatiana Nascimento. Trafegou em iniciativas dentro da Cultura Popular Tradicional utilizando a Música Eletrônica e a Produção Musical como ferramentas no tempo em que atuou na ação Cultura Digital do MinC, dentro do Programa Cultura Viva, até se envolver com a música experimental em 2013. Como resultado dessa perspectiva criou junto com o trompetista Mateus Aleluia Filho o duo Terra Vermelha. Hoje, dedica-se à sua pesquisa de Doutorado, onde desenvolve um trabalho que pensa nas interações da música da cultura popular tradicional com a música eletrônica a partir de uma perspectiva decolonial e contra-hegemônica.

SUBMERSO

(JOÃO MEIRELLES)

João Meirelles é compositor, produtor musical, live electronics performer e fotógrafo. Desenvolve desde 2010 seu trabalho autoral Infusão - projeto de performances musicais e composições através de improvisação. Desde 2012 faz live electronics na Baiana System, banda com impacto na cena da música independente brasileira e internacional. Compositor de diversas trilhas sonoras, João compõe e faz direção musical para artes do corpo, artes cênicas, cinema/animação e shows. Seus trabalhos recentes de Produção Musical são os álbuns "Olho de Vidro" de Jadsa e "Sistema Integrado Percussivo" de Japa System. Em 2020 lançou o Taxidermia Vol.1, projeto em parceria com a compositora Jadsa. Graduado em composição e mestre em interpretação e criação musical pela EMUS-UFBA, leciona atualmente cursos livres de produção musical.

FESTIVAL PANEL DISCUSSIONS / MESAS REDONDAS DA TRILHA DO FESTIVAL

Music, its creative practices and its resources/instruments advanced through technology and musical computing constitute an artistic ecology whose atmosphere is directly associated with environments and the will of individuals within this perspective.

Considering that there are topics within the musical creative universe that must be consistently present in all artistic dialogue environments, including within the context of the university and its academic and scientific events, SBCM 2021 invites all interested to the session of two panel talks that will take place on the 25th and 26th of October.

On both occasions, despite approaching apparently distinct subjects, they are not disjoint and represent an important initiative to dialogue important perspectives of musical practices in broad and inclusive spheres. On Panel 01, we will have the dialogue about “The use of technology in peripheral musical productions”. Then, panel 02 will be centered on the theme of “Gender in the musical computing spot”. Like it!

A música, as suas práticas criativas e os seus recursos / instrumentos desenvolvidos por intermédio da tecnologia e da computação musical constituem uma ecologia artística cuja atmosfera está diretamente associada com ambientes e os desejos dos indivíduos inseridos neste cenário.

Considerando que há temas dentro do universo criativo musical que devem estar presentes de forma consistente em todos os ambientes de diálogo artístico, inclusive, dentro do contexto da universidade e de seus eventos acadêmicos e científicos, o SBCM 2021 convida todos os interessados para a sessão de duas mesas de conversas que acontecerão nos dias 25 e 26 de outubro.

Nas duas oportunidades, apesar de abordar assuntos aparentemente distintos, não são disjuntos e representam uma importante iniciativa para dialogar perspectivas importantes das práticas musicais em esferas amplas e inclusivas. Na mesa 01, teremos o diálogo sobre “O uso da tecnologia nas produções musicais periféricas”. Já a mesa 02 será centrada no tema sobre “Gênero nos espaços da computação musical”. Curtam!

THE USE OF TECHNOLOGY IN PERIPHERAL MUSICAL PRODUCTIONS / O USO DA TECNOLOGIA NAS PRODUÇÕES MUSICAIS PERIFÉRICAS



CHICO CORREA

Chico Correa (Esmeraldo Marques), é um artista e produtor musical residente na Paraíba, cujo enfoque musical tem a música popular do nordeste em confluência com ritmos eletrônicos sintéticos como base. No seu estúdio produz trilhas sonoras para cinema e dança, além de remixes e experimentos com

processos criativos a partir de hardwares musicais.

Atuando como artista eletrônico, já circulou pelos festivais europeus: Фyзион (DE), Womex (HU), Babel Med(FR), Festival Mundial(NL), les Escales(FR), Europalia(BE), Med Festival (PT). Artista integrante do Label frances helicomusic.com.

Colaborador dos grupos Cabruera, Baiana System, Totonho e os Cabra, tem como trabalhos musicais o trio Berra Boi (@berraboi) , o duo com a Rapper Jessica Caitano @Surra_de_Rima e a banda Seu Pereira e Coletivo 401 (seupereiraecoletivo401).

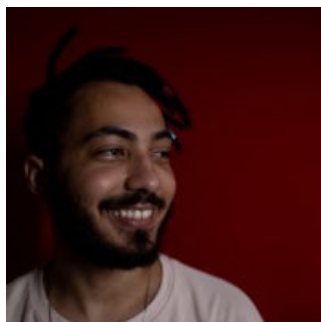


JOÃO MEIRELLES

João Meirelles é compositor, produtor musical, live electronics performer e fotógrafo. Desenvolve desde 2010 seu trabalho autoral Infusão - projeto de performances musicais e composições através de improvisação. Desde 2012 faz live electronics na Baiana System, banda com impacto na cena da música

independente brasileira e internacional. Compositor de diversas trilhas sonoras, João compõe e faz direção musical para artes do corpo, artes cênicas, cinema/animação e shows. Seus trabalhos recentes de Produção Musical são os álbuns “Olho de Vidro” de Jadsa e “Sistema Integrado Percussivo” de Japa System. Em 2020 lançou o Taxidermia Vol.1, projeto em parceria com a compositora Jadsa. Graduado em composição e mestre em interpretação e

criação musical pela EMUS-UFBA, leciona atualmente cursos livres de produção musical.



GG ALBUQUERQUE

Em 2020, foi curador da mostra musical Periferia Invenção, a ser realizada em março no Sesc Santo Amaro (adiada devido à pandemia). Escreveu e apresentou o documentário sobre bregafunk produzido pelo Spotify, em novembro de 2019. Foi jurado do Prêmio Multishow de Música Brasileira e curador do edital Natura Musical em 2020. Trabalhou como repórter de cultura do Jornal do Commercio e da Folha de Pernambuco e escreveu matérias sobre música de periferia para o Portal Kondzilla, UOL Tab, Vice Brasil, Bandcamp, Revista Continente, Revista Noize, Monkeybuzz, entre outros.



THIAGSON

Thiagson, pesquisador do funk, é bacharel em Composição Musical pela Unesp e doutorando em Musicologia pelo Departamento de Música da USP. Em sua dissertação de mestrado, abordou o debate sobre a legitimidade do funk na universidade e nas redes sociais. É autor dos livros “Sorry It’s Over: A Morte da Música Clássica” (2018) e “Apontamentos sobre Alban Berg” (2013). Desenvolve trabalho de divulgação científica nas redes sociais e YouTube, por meio do Canal do Thiagson, além de ministrar palestras sobre funk e música.

GENDER IN THE MUSIC COMPOSITION SPOT / GÊNERO NOS ESPAÇOS DA COMPOSIÇÃO MUSICAL



ANA MARÍA ROMANO G.

Compositora, artista sonora e interdisciplinar colombiana. Sus intereses investigativos se sitúan desde la intersección entre género, sexualidades, sonido y tecnología, atravesados por la escucha, el paisaje sonoro, el espacio, el cuerpo, el noise, la experimentación, el ciberespacio y las dimensiones políticas en la creación. En todos los espacios de su vida considera fundamental el trabajo colectivo y colaborativo.



ARIANE STOLFI

Ariane Stolfi é arquiteta, designer, programadora, musicista e professora, e transita por várias linguagens e disciplinas. Doutora em Sonologia pelo departamento de Música da Universidade de São Paulo (ECA-USP) e Mestre em Arquitetura e Design (FAU-USP), pesquisa interfaces interativas em tecnologias web, atualmente na Universidade Federal do Sul da Bahia. Foi pesquisadora visitante no Centre for Digital Music (Queen Mary University of London) e participou de diversos festivais e eventos como: Festival de Música Nova, Digitália, SHA, Submidialogias, DIS Experimental, ¿Música?, Virada Cultural, Bigorna, Improvise!, Circuito de Improvisação Livre, Amostra Sonora, Áudio Insurgência, Tecnoxamanismo, CMMR, Web Audio Conference, Audio Mostly, UbiMus e NIME. Contato: arianestolfi@gmail.com.



TIAGO IVE RUBINI

Tiago Ive Rubini trabalha com arte e pesquisa. Atualmente faz doutorado no Instituto de Artes da Universidade Estadual Paulista Júlio de Mesquita Filho (UNESP) e trabalha na Academy of Media Arts Cologne, na Alemanha. Se interessa por arte sonora e processos artísticos de código aberto, principalmente no âmbito do conhecimento compartilhado e ativismo interseccional. Fez graduação em Estudos de Mídia pela Universidade Federal Fluminense (UFF) e mestrado em Artes Visuais, Música e Tecnologia pela Universidade Federal de Juiz de Fora (UFJF). Participou de exposições em eventos e instituições como a Bienal de Curitiba, a Escola de Artes Visuais do Parque Lage, no Rio de Janeiro, e o Centro de las Artes de Sevilla, na Espanha.

Contents

Scientific Papers

Automatic classification of instruments from supervised methods of machine learning	1
Rômulo Vieira, João Teixeira Araújo, Edmilson Batista dos Santos, Flávio Schiavoni	
Instance Selection for Music Genre Classification using Heterogeneous Networks	8
Angelo Cesar Mendes Silva, Paulo do Carmo, Diego Furtado Silva, Ricardo Marcondes Marcacini	
Tempo estimation via neural networks a comparative analysis	17
Mila Soares de Oliveira de Souza, Pedro Nuno de Souza Moura, Jean-Pierre Briot	
An interplay between genre and emotion prediction in music: a study in the Emotify dataset	25
Leonardo Vilela de Abreu Silva Pereira, Tiago Fernandes Tavares	
Mandrit: a system for automatic generation of musical visualizations for rhythm analysis	30
Rute Moura, Horhanna Almeida de Oliveira, Giordano Ribeiro Eulálio Cabral, Ricardo Scholz	
On the Fusion of Multiple Audio Representations for Music Genre Classification	37
Diego Furtado Silva, Micael Valterlânio da Silva, Ricardo Szram Filho, Angelo Cesar Mendes Silva	
Towards Realtime Score Analysis in PWGL	45
Mika Kuuskankare	
Evaluating the Automatic Chord Estimation and Alignments tasks needs using metrics from a code challenge	50
Valter Jorge da Silva, Giordano Ribeiro Eulálio Cabral	
Training a convolutional neural network for note onset detection on the clarinet	55
Tairone N. Magalhães, Maurício A. Loureiro	
On Generalist and Domain-Specific Music Classification Models and Their Impacts on Brazilian Music Genre Recognition	60
Diego Furtado Silva, Angelo Cesar Mendes Silva, Luís Felipe Ortolan, Ricardo Marcondes Marcacini	
Applications of FFT for timbral characterization in woodwind instruments	68
Yubiry González, Ronaldo C. Prati	
Real-time Qualification of Percussive Sounds Based on Correspondences Between Schaeffer's Solfège and Low-level Audio Descriptors	74
Sérgio Freire, José Padovani, Caio Campos	
A Roughness Model Implementation to Analyze Sound Mass Composition: A Study Case in Ligeti's Continuum for harpsichord (1968)	82
Micael Antunes da Silva, Guilherme Feulo do Espírito Santo, Jônatas Manzolli	
Concatenative Sound Synthesis as a Technomorphic Model in Computer-Aided Composition	88
Júlio Guatimosim, José Padovani, Carlos Guatimosim	
grainBirds: Evolutionary granular synthesisers distributed in Fog Computing	95
Jônatas Manzolli, Edelson Constantino	
Correlating Big Five Primary Personality Dimensions with Musical Preferences	103
Luiz Delando Santos Moreira Júnior, Guilherme Ribeiro Eulálio Cabral, Giordano Ribeiro Eulálio Cabral	
Motus 4: new developments on a web-deployed software for composing real-time audiovisual interactions.	109
Augusto Piato Amstalden, Tiago Fernandes Tavares	
Design process and rapid prototyping of animated music visualizations	114
Horhanna Almeida de Oliveira, Giordano Ribeiro Eulálio Cabral, Rute Moura	
Electric guitar distortion effects unit using a Raspberry Pi	121
Renato Pereira, Rodrigo Andreão	
Challenges to generate musical visualizations	129
Rute Moura, Giordano Ribeiro Eulálio Cabral, Jäder Abreu, Mychelline Cunha, Horhanna Almeida de Oliveira	
A preliminary study of Augmented Musical Instruments for Study (AMIS) using research through design.	136
Eduardo Santos Silva, Giordano Ribeiro Eulálio Cabral, Rodrigo Mendes de Carvalho Pereira	
CatalogToMakers: a collaborative cataloging platform for electronic components and physical computing projects related to musical creativity	144
Michael L. Bastos, Filipe Calegario, Giordano Ribeiro Eulálio Cabral	

Hot Streaks in the Brazilian Music Market: A Comparison Between Physical and Digital Eras	
Gabriel R. G. Barbosa, Bruna C. Melo, Gabriel P. Oliveira, Mariana O. Silva, Danilo B. Seufftelli, Mirella M. Moro	152
The 4E's Model of Enactivism through Improvisation within DMLs	
Isabela Corintha, Giordano Ribeiro Eulalio Cabral	160
An open source platform to assist the creation of group playlists through artificial intelligence algorithms	
Flaviano Dias Fontes, Giordano Ribeiro Eulalio Cabral, Geber Lisboa Ramalho	166

Art Papers

Playing Time-Variant Audio Feedback Networks	
Adriano Monteiro	172
Lick the Toad: a web-based interface for collective sonification	
Konstantinos Vasilakos	178
Mapping, Triggering, Scoring, and Procedural Paradigms of Machine Listening Application in Live-Electronics Compositions	
Vinicius César Oliveira, José Henrique Padovani	189
Acalma-te: resultados de composição automática e iterativa para relaxamento	
Cláudio Gomes	197

Short Papers

Investigando Tecnologias Educacionais para o Ensino e Aprendizagem de Música para Deficientes Visuais Totais	
Sara Elene Lobato dos Remedios, Sandro Luis Freire de Castro Silva, Simone Bacellar Leal Ferreira	201
Production of digital content in music teacher education: a study about podcast's possibilities	
Marcos da Rosa Garcia, Gutenberg Marques, Matheus Barros, Juciane Araldi Beltrame	205
Song Emotion Recognition: A Short Survey	
Arthur Nicholas dos Santos, Karen Gissell Rosero Jácome, Bruno Sanches Masiero	209
Análise de características acústicas para identificação de triggers em gêneros e áudios musicais distintos	
Marco Antonio Santos Fernandes, Estela Ribeiro, Carlos Eduardo Thomaz	213
Identificação de Áudio Vocaloid com Support Vector Machines: um Estudo de Caso da Hatsune Miku	
Felipe V. de Almeida, Victor T. Hayashi	217
Convolutional Neural Network applied to Identification of Musical Elements in OMR	
Jenaro Augusto Barbosa, Edmilson Batista dos Santos	221
Modelo automático de classificação de gêneros musicais amazônicos	
Douglas Sousa, Lucas Zampar, Felipe Rodrigues, Cláudio Gomes	225
Camomile-ELSE: creating plugins with Pure Data patches and the ELSE library	
Esteban Viveros Astorga, Alexandre Torres Porres	229
Web Art em tempos de HTML5	
Avner Maximiliano Paulo, Luiz Gustavo Colzani Monti Sousa, José Mauro da Silva, Flávio Schiavoni	233
Cerâmica e silício na criação musical	
Carlos Eduardo Oliveira de Souza, Rafael Alves Soares de Andrade, Jônatas Araújo da Silva, Flávio Schiavoni	237
Rapid prototyping: using Wizard of Oz to emulate machine learning features for interactive artistic applications	
Alessandra Aleluia, Giordano Cabral	241
Expressividade de instrumentos musicais digitais - just push play	
Gabriel Lopes Rocha, Flávio Schiavoni	245
Desafios da Internet das Coisas Musicais	
Alexandre Silva de Almeida, Rômulo Vieira, Rafael Sachetto Oliveira, Flávio Schiavoni	249
CongAr: Instrumento digital musical de baixo custo para internet das coisas musicais	
Douglas Silva, Josemar Gama, Joelden Lopes, Lucas Zampar, Felipe Rodrigues, Cláudio Gomes	253

Lab Reports

A prospective report on the research developed at the Laboratory of Audio and Music Technology at USP	
Regis Rossi A. Faria, Ricardo Thomasi, João Monnazzi, Eduardo Bonachela, Andre Giolito, Gabriel Lemos	259

Alice no país da pandemia (2021)

Flávio Schiavoni, Ana Clara Medina Fonseca, Avner Maximiliano de Paulo, Carlos Eduardo Oliveira de Souza, Cleisson José Dias da Silva, Emerson Junio Silva Costa, Fábio dos Passos Carvalho, Gabriel Lopes Rocha, Gabriel Rodrigues Chaves Carneiro, Igino de Oliveira Silva Júnior, Igor Alves Estevam Andrade, Isadora Franco Oliveira, João Marcos de Oliveira Lara, João Teixeira Araújo, João Pedro Mendes de Oliveira, Jônatas Araújo da Silva, Luan Luiz Gonçalves, Lucas Estevão dos Santos, Luiz Gustavo Colzani Monti Sousa, Mariana Pereira Lellis, Matheus de Bonfim Rodrigues Jordão, Rafael Alves Soares de Andrade, Rebeca Lima Soares, Rômulo Augusto Vieira Costa, Samuel Rodrigues Rabay	266

CEGeME 2021 - Gestos, Movimentos, Performance musical e Pandemia

Maurício Alves Loureiro, Alexandre Silva, Thiago de Almeida Magalhães Campolina, Flávio Schiavoni, Davi Alves Mota, Aluizio Barbosa de Oliveira Neto, Tairone Nunes Magalhães, Tiago Lima Bicalho Cruz, Gustavo Machado Oliveira, Felipe Brandão Barros, Luis Fernando Umbelino da Silva, Ravi Shankar Viana Domingues, Renato Rodrigues Lisboa	272

COUT-e e práticas itinerantes durante 2020 e 2021

Douglas Sousa, Lucas Zampar, Felipe Rodrigues, Cláudio Gomes	280

Relatório de Pesquisa NESCoM 2021

Leandro Costalonga, Marcus Vinícius Neves	282

Workshops

Introdução à computação musical

Cláudio Gomes	289

Oficina Música Visual gerativa

Rute Moura, Jarbas Jácome, Vinícius Garcia, Gustavo Carneiro, Horhanna Almeida, Mavi Pugliesi	293

Towards a Latin American NIME Research Community

Juan Pablo Martínez Avila, João Tragtenberg, Filipe Calegario, Juan Ramos, Martin Matus Lerner, Isabela Corintha, Javier Jaimovich, Teodoro Dannemann, Patricia Cadavid, Ximena Alarcón, Adnan Marquez-Borbon, Hugo Solís, Miguel Ortiz	298

Demos

Música Móvel - Criação sonora para web

Cristiano Figueiró, Letícia Mayni, Leonardo Almeida	304

ContraMonumentos | CounterMonuments

Andreia Oliveira, Davi Carvalho, Tulio Chiodi, Matheus dos Santos Camargo, Wagner Antonio, Camila dos Santos, Cristiano Figueiró	306

Index

Authors	309
Institutions	310
Keywords	311

Scientific Papers

Automatic classification of instruments from supervised methods of machine learning

Rômulo Vieira^{1,2}, João Teixeira Araújo^{1,2}, Edmilson Batista², Flávio Schiavoni^{1,2}

¹Arts Lab in Interfaces, Computers, and Everything Else - ALICE

²Department of Computer Science - UFSJ

romulo.vieira96@yahoo.com.br, teixeira.araujo@gmail.com, edmilson.batista@gmail.com, fls@ufsj.edu.br

Abstract. *Sorting instruments is not an easy task for humans or computers, especially when it comes to elements with the same acoustic properties, such as wind, percussion, or strings. Nevertheless, the use of audio descriptors and artificial intelligence techniques can make this duty more accessible. In this paper, three supervised methods, Naive Bayes, decision tree and Support Vector Classifier (SVC) are used to categorize acoustic guitar and bass sounds in a database, using as a parameter the information extracted from audio descriptors. The research resulted in a performance comparison of these three algorithms, considering their hit rates and processing time when classifying samples in different parts of the dataset. After all, some relevant considerations about the feasibility of automatically classifying instruments are presented.*

1. Introduction

The constant technological advancement has enabled several changes in the area of computer music, which facilitates the creation of various works in digital format. Given the vast amount of generated files, there is a need to classify them based on their main characteristics. One way to do this is through metadata, where tags defined by experts and consumers inform the author, album, style, year of release, or record label of a sample.

However, this classification method presents some problems, such as high cost to hire professionals capable of performing this categorization properly; slow process, as it is manual; divergences in classifications used by different groups and users; preference for famous artists, which overshadows those more unknown; inability to fully capture the musical and cultural content of the tracks; and finally, the difficulty for the lay public to associate the true meaning of each tag [1].

Consequently, the automation of this process becomes an area of interest to provide better organization and recommendation of this content, especially when dealing with a very large database. In this context, the present paper uses supervised methods, audio descriptors, and a multi-classifier algorithm to separate the samples of a database between acoustic guitar and bass, based on the timbre of each one of them. Then, comparisons and analyzes are performed on the processing times of each method and the feasibility of automatically classifying instruments.

The remainder of the paper is structured as follows: Section 2 indicates the related works and the state of the art in classifying instruments by digital means; Section 3 presents a brief analysis of digital audio and the descriptors used, while Section 4 shows the methodology and applied tools in the classification. Section 5 displays the obtained results and a discussion about them. Finally, Section 6 brings summarized conclusions of this work.

2. Related Works

The classification of musical properties is a popular task among artificial intelligence researchers so that there is a multitude of papers dealing with this topic, using the most diverse techniques. A work that comes very close to what is proposed throughout this text is that of Lara Haidar-Ahmad [2], who applies a multi-classifier in an audio stream to classify the timbres into drums, piano, flute, or “other” when none of the above are identified. The audio is preprocessed to extract a Mel-spectrogram and the outputs determine the dominance or non-dominance of a given instrument in that analyzed audio track.

Other works use Support Vector Machines (SVMs) and Gaussian models to solve this problem, as can be seen in [3] and [4]. The first article classifies 8 instruments using two Gaussian methods and SVM, reaching an accuracy of 70%, while the second combined the Gaussian technique with a k-nn classifier to group instruments based on their features.

More recently, researchers have turned their attention to deep learning in performing this task, which removes the manual extraction of features from audio samples. From this, models emerged that can learn from the frequency spectrum of these same samples [5].

In the end, our literature research showed that most studies use a dataset with isolated sounds, consisting of the presence of a single instrument per sample. The same pattern was repeated in this work, with the difference that here, a much larger amount of samples was used to infer results. Another difference is that our work focus was on supervised methods, while the state of the art prioritizes unsupervised learning techniques.

3. Sound Analysis

Unlike analog audio, digital sound is not continuously represented, requiring conversion from one format to another.

This is possible thanks to a subfield of Computer Science, called signal processing, which performs the electronic manipulation of acoustic data received as input [6].

Another technology that helps in this process is the audio descriptor, an analytical tool that represents the characteristics of the musical signal in a dimensional curve. These descriptors reduce the complexity of information by focusing on specific aspects of the signal. They are still useful for creating a particular taxonomy of the content of the musical signal spectrum because they have a reductionist character. These attributes can be correlated with subjective perceptual properties related to timbre, such as “brightness”, “opacity” or even “smoothness” of the sound, whereas timbre is defined as a result of the combination of two components: the vibrations of the sound and the frequencies produced by these vibrations. It is important to say that each source produces sounds in different ways, which can be by fingering, percussion, breathing or electronic inference [7, 8].

In this paper, four descriptors will be used: Spectral Rolloff, Spectral Centroid, Spectral Flatness, and Mel-Frequency Cepstral Coefficient (MFCC). The first one extracts the sample’s rollover point, that is, the segments of the sound wave that are below a pre-defined percentage. The second model, as far as it is concerned, is responsible for indicating where the center of mass of the spectrum is, which represents the central tendency of the waveform. It has a connection with the impression of “brightness” of the sound, and its most common applications are aimed precisely at timbre classification.

The third descriptor is also known as Wiener Entropy. It is a measure used to quantify the appearance of noise as opposed to pitching. An example of how this tool works can be seen in Figure 1 [9, 10].

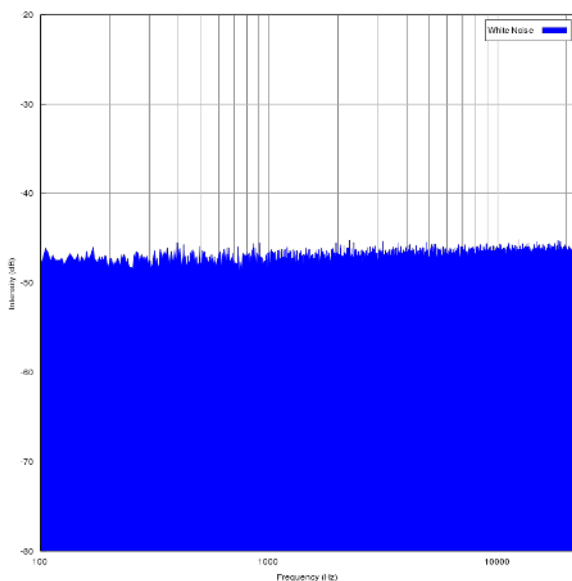


Figure 1: Flatness descriptor spectral curve [11].

The fourth and last descriptor is also the most important. It is a derivative of the representation of a non-

linear audio spectrum, widely used in speech recognition systems, musical genre classification, and audio similarity measures. The fact that the MFCC is not robust to noise is also noteworthy, therefore, its values must be normalized. Figure 2 shows an audio sample after applying this descriptor [12].

4. Methodology

To carry out this study, we use some tools and libraries available for the Python programming language. One of these libraries was Librosa¹, developed for applications that involve some type of digital audio processing, to facilitate work involving the recovery of musical information. Among its advantages is familiarity with MATLAB, standardization of commands, and modular features, in addition to features that allow you to estimate the beats per minute in certain samples and the graphic representation of the sound in spectrograms. These fundamental properties can be observed in the Figure 3 [13].

Another library used to handle with audio was AudioLib². Although less powerful than the Librosa, it provides a high quality system for reproducing sound data, its main feature being the ability to generate waveforms, which is essential for timbre differentiation.

Of the libraries for data classification, the chosen one was scikit-learn³, as known as Sklearn, widely used to identify the category and continuous value of a given object, as well as automate the grouping of family values and extract and normalize features. It also allows integration with well-established technologies in the field of Artificial Intelligence, such as NumPy, Matplotlib, and Pandas [14].

The database that provided the information to be classified is Nsynth, considered a reference in the study of audio synthesis that presents large magnitudes, with about 20 GB and 305,979 musical notes, where each one has a unique timbre, pitch and envelope. These notes can be

¹<https://librosa.org/doc/latest/index.html>

²<https://pypi.org/project/AudioLib/>

³<https://scikit-learn.org/stable/index.html>

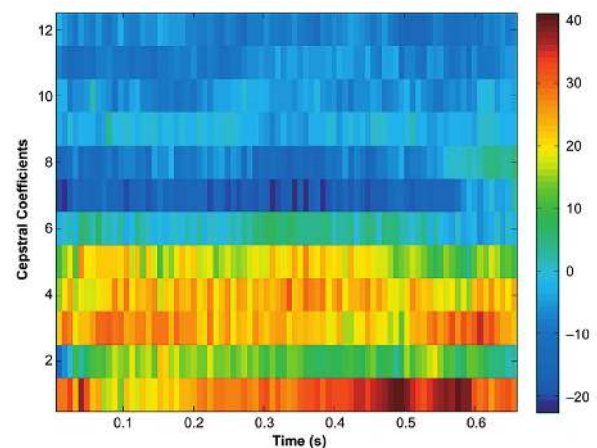


Figure 2: Mel-Frequency Cepstral Coefficient spectrogram [12].

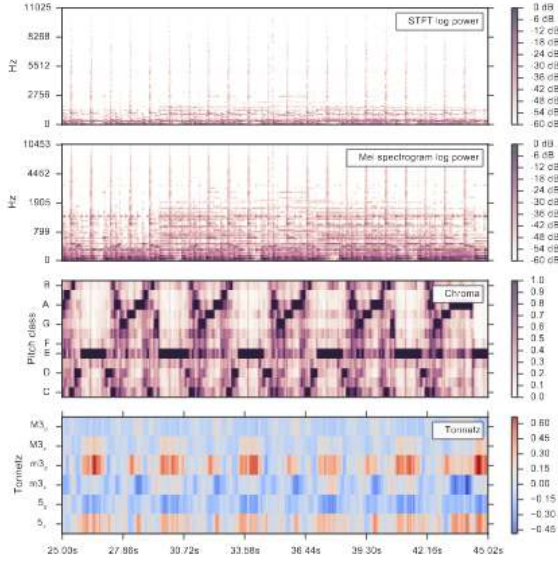


Figure 3: Spectrograms generated by the Librosa library

further divided according to their font, family and quality. The first category refers to the method of sound production (acoustic, electronic or synthesized), while the second category is responsible for classifying the instruments according to their main characteristics and the third separates the samples based on their sound resolution [15].

Despite being robust, NSynth has some particular characteristics that can interfere with the classification of instruments. Starting with the way the samples were created. Regardless of whether they represent acoustic guitar, bass, flute, or another instrument, their means of production was a standard MIDI piano. In consequence, tones range from 21 to 108 and detected speeds are 25, 50, 75, 100, 127. Therefore, not all instruments are capable of producing the 88 tones of a normal range, resulting in an average of 65.4 tones per instrument. There are also sample packs that contain sounds duplicated at different speeds, resulting in a unique 4.75 speed per pitch. These characteristics directly interfere in the investigation of the timbre, resulting in guitar samples with notes as low as a bass and bass notes as high as a guitar note [16].

4.1. Classification Algorithms

The process of classifying sound samples can be performed by applying machine learning techniques. That is, in a pre-processed database, the computer follows a set of steps to learn to differentiate between two different audios. One of the most famous methods for this function is the Naive Bayes algorithm, a probabilistic classifier, based on the theorem of the mathematician Thomas Bayes, which lends your name to the tool. This theorem consists of multiplying a probability collected before the system is executed (*a priori*) by a probability obtained after the occurrence of the event (*a posteriori*). This technique is based on conditional independence of values, where the presence of a certain event has no correlation with any other, hence

the name “Naive Bayes”. Its mathematical representation can be seen in the equation 1 [17].

$$P(A | B) = \frac{P(B | A)P(A)}{P(B)} \quad (1)$$

$P(A)$ and $P(B)$ are the *priori* probabilities of their respective events. $P(A|B)$ is the *posteriori* probability of A, after the occurrence of B event, while $P(B|A)$ is the occurrence of B after A. Its main advantage is that it only needs a few training data to estimate the values. It is considered one of the most effective methods for dealing with real-world problems, although its construction is relatively simple.

Another form of classification is using decision trees. This is a hierarchical structure characterized by supervised learning. As a consequence, the input data has its characteristics defined from the beginning and uses this knowledge to act on the new data that is received [18].

These trees are formed by nodes, which represent data attributes, by branches, which indicate the classification rules, and by leaves, which refer to the results. The objective of the algorithm is to progressively divide the database, partitioning the space into sub-regions based on a descriptive feature. This space reduction happens until they are so small that they can be classified using a single label. From there, the new input data are also categorized in this way.

For the problem discussed here, suppose a new audio sample has been inserted into the system and must be designated as a bass or acoustic guitar. The first question the algorithm must answer is how often this sample matches the root node of the decision tree. From there, other questions can be asked based on the previous answer. This sequence is organized to generate a tree, which will indicate the way forward to arrive at a ranking. For this, the algorithm simply leaves the root node of the tree and crosses it to a leaf node, responsible for indicating the class of that audio. Each choice for a different segment will result in different results.

Some other information is important for building a tree. The first one to be mentioned is the gain information, which can be seen in the equation 2 and represents what was learned about the labels when a region is divided. In this context, H indicates the impurity of the region R, while R_e is the left subregion and R_d the right subregion.

$$Gain(R, R_e, R_d) = H(R) - \frac{|R_e| * H(R_e) + |R_d| * H(R_d)}{|R|} \quad (2)$$

The second prominent factor is entropy, responsible for indicating the degree of purity and instability of the classifier. For this, it resorts to parents and child nodes, where the attribute that generates the greatest difference will be chosen. This condition can be seen in the equation 3 [19].

$$\text{entropy}(R) = - \sum p(c|R) \log(p(c|R)) \quad (3)$$

$P(c|R)$ indicates the probability that a point in region R belongs to class c . This probability is estimated by the ratio between the number of these points.

The last point to be considered when planning this method is the Gini factor, observed in the equation 4. For this condition, the variables c and R assume the same values as in the entropy calculation, with the highest index generated being the one chosen to integrate the system [19].

$$\text{Gini}(R) = \sum p(c|R)(1 - p(c|R)) \quad (4)$$

This paper also made use of two more classification methods: Support Vector Classifier (SVC) and multi-classifiers. The SVC is based on an adjustment of the data provided, applying penalties, loss functions, metric scores, and decision limits to determine which one best fits the analyzed problem. The best performing data is then transported to a hyperplane, where it will be fed some enhancement features to decide the classes of the input objects. Although it has some disadvantages, such as not punishing interception and converging at a slower time compared to other techniques, it was chosen because of its ability to be an individual scheme. In this way, each classifier is directly compared to another classifier, showing better scalability of penalty choices in large amounts of samples.

Multi-classifiers, on the other hand, consist of combining and applying a set of classification algorithms in the parts of the input space where they have better performance, in order to optimize the output values. This category of the system is characterized by allowing weak data to provide good results and not requiring such a precise adjustment of the base. In the classification of bass and acoustic guitar samples, the multi-classifier used was Bagging. This procedure consists of choosing a random sample from the dataset and generating new subsets from it. In each of these divisions, a different classification is applied, so that the algorithm obtains diversity in the response models. At the end of this step, it starts a simple voting process to determine which class to choose. The selected class is the one that got the best answers for the different classifiers. It is noteworthy that in this work the training and testing bases were applied equally to the 3 classifiers, not a subset for each one, as is the default in Bagging [20].

It is important to highlight that in a classification problem, data mining plays a role as important as the classification system itself. Good data selection prevents overfitting, improves performance, and reduces training time. A tool that helps in this task is the F-test, commonly used to better identify the model that best fits a certain population. For this, each test receives an important value for each resource, according to the improvement or deterioration caused in the system. For the problem proposed here,

the F-test was used to find the 3, 5, 6, and 8 best samples [21].

4.2. Pre Processing Data

For this stage, 2500 bass and acoustic guitar samples were initially collected, at different frequencies. Through the Librosa library, the numerical values of each sample were extracted, as an audio/sound wave can be represented by a vector, which are used to reconstruct a sound wave from its sampling rate and the application of trigonometric equations.

Using this vector as a basis, the Short-Time Fourier Transform (STFT) was applied. This technique consists of dividing the spectrogram into smaller intervals, making each one of them become a constant. Subsequently, a Fourier transform is applied to each of these partitions. The method was used to collect the modulus of the vector result, as it has negative and positive values.

After this procedure, the STFT output is used in the audio descriptors to provide another numerical vector having specific audio characteristics. From this new vector, the mean and standard deviation between each one is applied, providing two features per descriptor.

In this way, each sample is treated and ready to be used in classification. Regarding the pre-processed data, there are 8 features for each sample. Furthermore, the class referring to the sample is returned, which can be 0 for bass and 1 for acoustic guitar. No problems were found regarding inconsistent or missing data.

5. Results and Discussions

We conducted our experiments separated in three steps, using different partitions of the dataset, namely: i) partial use of NSynth, containing 5000 guitar and bass samples; ii) full use of NSynth, containing all 24123 guitar and bass samples; and iii) full use of NSynth with all electronic guitar and bass samples, reaching a total of 99659 copies. In each of the three steps, the previously mentioned algorithms were used. It is noteworthy that all algorithms were executed in the f1 test and the number of resources used was gradually increased, generating for each stage, five tests based on the 2, 3, 5, 6 and 8 main features.

5.1. Step 1: Partial NSynth dataset

Initially, 5000 samples were collected, 2500 referring to acoustic guitar and 2500 to bass. Despite providing acoustical and synthetical samples, it used only electrical ones, with a database randomly divided between 60% for training and 40% for test.

Even with a small base, the algorithms demonstrated different types of behavior. In relation to Naive Bayes and the algorithm provided by Sklearn, by using the same technique for classification, they obtained an almost identical result, starting in the range of 69% for 2 features, and increasing the precision significantly until reaching 6 features, where it stabilized at 80%.

The decision tree was the algorithm that achieved the best accuracy, reaching a peak of 92% with 5 features and declining slightly in the following analyses. SVC, else ways, presented a very peculiar behavior. Under the use of up to 3 features, he got a result of 75% hit, however, the increase in the number of features caused a considerable drop of almost 25%. This behavior highlights the idea that increasing features does not always improve the accuracy of an algorithm.

On the implemented multiclassifier, the technique used was the voting system, where the Naive Bayes, decision tree and SVC algorithms were used. In this way, each instance received classifications from the algorithms, with the most frequent class being the one used as a result. It can be seen that, based on the voting system, the multi-classifier ends up tending to behave similarly to the best classifier used. In this case, even with worse accuracy, the multi-classifier obtained results very similar to those of the decision tree. It is noteworthy that the decrease in SVC accuracy from the use of 5 features did not affect the results of this method. All these behaviors are summarized in Figure 4.

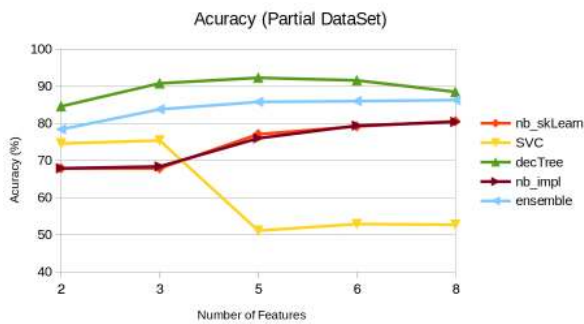


Figure 4: Partial Dataset Classification.

Aiming at a deeper analysis of the influence of the descriptors on the result of the classifiers, the correlation matrix between the invoices was created, which can be seen in Figure 5.

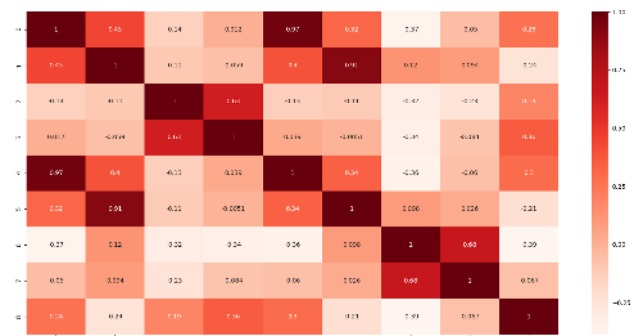


Figure 5: Correlation Matrix Between Features.

As much as the matrix demonstrates all the correlations between the resources, the task of defining which ones were really important is a little difficult. For this, taking into account the classes of each sample, a ranking was made of which characteristics were more correlated with

each other. From the generated classification, it was noticed that the use of the mean and standard deviation under a descriptor does not always result in features that will have similar correlations in relation to the classes. In Figure 6, it is observed that the four most correlated features were collected from different descriptors. That is, when the mean for a descriptor resulted in a similar feature, the standard deviation resulted in an inverse behavior.

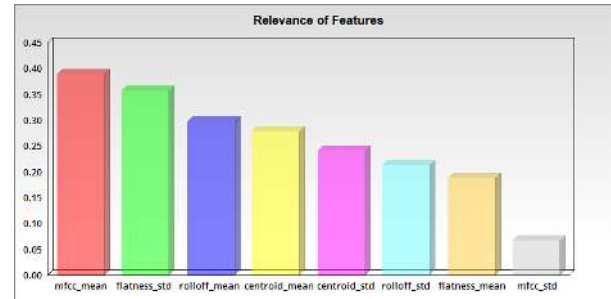


Figure 6: Feature Ranking.

The correlation between features is not always directly related to the accuracy of the classifier. Thus and so, removing a feature that has a lower relationship to the others will not necessarily cause an improvement in accuracy.

Regarding the execution time for this step, in addition to the SVC that obtained times between 400 and 600 milliseconds, the other algorithms obtained acceptable times, ranging between 2 and 30 milliseconds.

5.2. Step 2: Full NSynth dataset

The results of the first stage were obtained from a very small and specific database. Wherefore, step 2 used the complete database, totaling 99659 samples, being 33342 guitar and 66317 bass, available in acoustic, synthetic and electric formats. In respect of training and testing data, Nsynth already provides the partitioned data, totaling 98164 samples for training and 1495 for testing.

While the results of step 1 averaged between 75% and 95%, when using the entire database, a better result was expected, since the use of more input values improves the training of classifiers, and consequently, increases accuracy. However, the results ended up being between 50% and 65%, as shown in Figure 7.

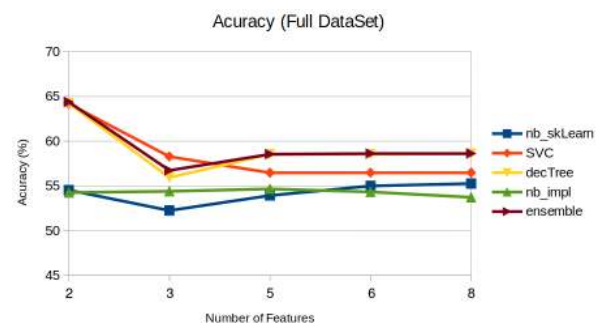


Figure 7: Full Dataset Classification.

In other words, even with the increase in the database, some other factor ended up causing a sharp drop

in the accuracy of the algorithms. As the database used ended up encompassing samples from more sound sources and not just electric instruments, the timbre classification may have been compromised.

In relation to the execution time of the algorithms, the SVC had a much higher execution time when compared to the others. Despite its contribution to the design and its use being desirable to reduce calculation time, the basic composition of the SVC made it have worse performance. This is because the kernel acts as a space function, which can be highly complex. It is also necessary to consider the size of the database to which the SVC was submitted, as this can cause the kernel array to be stored directly in memory, thus affecting its performance as a whole. Still, on the performance of the algorithms in relation to the execution time, all the algorithms had their execution worsened by increasing the number of features desired. Figure 8 shows this decay.

time (ms)	nb_skLearn	SVC	decTree	nb_impl
2 features	17.12	551539.33	200.41	285.21
3 features	17.06	493275.20	289.28	344.80
5 features	19.30	589837.13	456.41	396.59
6 features	20.02	624353.81	550.10	507.95
8 features	21.76	638291.16	755.98	669.39

Figure 8: Full Dataset Time Classification.

5.3. Step 3: Full electronic NSynth dataset

To better investigate the reason for the accuracy decrease, a third and final analysis was applied, but now only with electronic instruments, totaling 16206 guitar samples and 7917 bass samples. Besides, it is possible to investigate whether the drop in accuracy is related to samples generated by different sound sources. Once the tests were carried out in this way, the algorithms returned to obtain good results, with an accuracy between 65% and 95%, without considering the SVC result, which was much lower than the other classifiers. Such values are graphically represented in Figure 9.

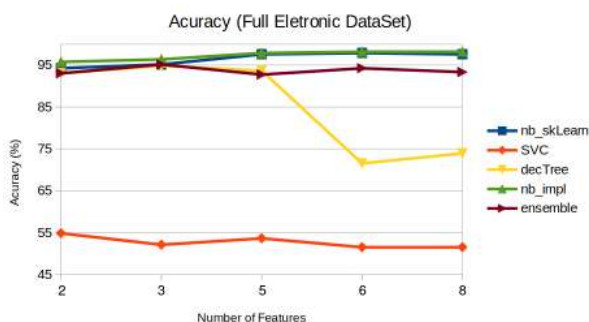


Figure 9: Full Electronic Dataset Classification.

Analyzing each algorithm individually, the decision tree obtained a decrease in precision when using 6 and 8 features. Naive Bayes achieved the best results, even being better than the multi-classifier. This fact can be explained by the behavior of the SVC, which obtained low accuracy, between 50% and 55%. That is, the SVC ended

up casting wrong votes for the multi-classifier, considerably reducing its accuracy. Consequently, we conclude that implementing a voting system may sometimes not be a good choice when one of the classifiers used is getting poor results.

It is also observed that the execution time of the algorithms remained acceptable, below 200 milliseconds. The SVC, oppositely, because it uses complex functions, ended up obtaining an execution time between 29 and 35 milliseconds.

6. Conclusions

This paper presented an automatic instrument classification experiment using a well-known dataset, audio descriptors commonly used to identify timbres, and three different classification algorithms: Naive Bayes, Decision Tree, and SVC.

From the analysis of the results, it is clear the importance and reason for choosing each of these algorithms. Naive Bayes stands out for being very efficient in estimating probabilities, having a simple and easy-to-implement structure. The decision tree has the advantages of easy implementation and interpretation of data compared to more traditional models, while SVC appears as an alternative for the part of the database that was not as robust or had some noise.

Finally, it is noteworthy that the complexity of the problem of dividing two timbres is strongly influenced by the categories of instruments used. The task of distinguishing a drum from a flute, for example, can be easier than differentiating string instruments with similar timbres. Hence, the importance and applicability of automatic classification systems for music are explicit. Another strong point to be highlighted is the interdisciplinary character of this field of action, assisting in research, projects, and teaching of the most diverse types of science.

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 (Grant Number 151975/2019-1), CAPES (Grant Number 88887.486097/2020-00) and FAPEMIG.

References

- [1] Roberto Piassi Passos Bodo and Marcelo Gomes de Queiroz. Três abordagens para similaridade musical utilizando melodia, ritmo e timbre. Master's thesis, Universidade Federal de São Paulo, São Paulo, 2018.
- [2] Lara Haidar-Ahmad. Music and instrument classification using deep learning technics. In *Proceedings of the Conference on Neural Information Processing Systems*, pages 1–6. Nips.cc, 2017.
- [3] Janet Marques and Pedro Moreno. A study of musical instrument classification using gaussian mixture models and support vector machines. *Cambridge Research Laboratory - Technical Report Series*, 09 1999.

- [4] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.00CH37100)*, volume 2, pages II753–II756 vol.2, 2000.
- [5] Babak Toghiani-Rizi and Marcus Windmark. Musical instrument recognition using their distinctive characteristics in artificial neural networks. *ArXiv*, abs/1705.04971, 2017.
- [6] Martin Vetterli Paolo Prandoni. *Signal Processing for Communications (Communication and Information Sciences)*. EPFL Press, 2008.
- [7] Ivan Eiji Simurra. A utilização descritores de áudio à análise e composição musical assistidas por computador: um estudo de caso na obra labori ruinae, 2016.
- [8] Ivan Eiji Simurra. Análise musical assistida por descritores de Áudio: um estudo de caso da obra reflexões de jônatas manzoli, 2015.
- [9] Sound Analysis Pro Editorial. Wiener entropy, Outubro 2014. [Online; chapter 4].
- [10] Lusciniã. Spectrogram window, Outubro 2017. [Online; Parameters].
- [11] Geoffroy Peeters. A large set of audio features for sound description. *Ircam - Analysis/Synthesis Team*, 2004.
- [12] N. Fakotakis e G. Kokkinakis T. Ganchev. Comparative evaluation of various mfcc implementations on the speaker verification task. *10th International Conference on Speech and Computer (SPECOM 2005)*, 2005.
- [13] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. *librosa: Audio and music signal analysis in python. 14th Python in Science Conference, s/n(s/n)*, 2015.
- [14] Eli Bressert. *SciPy and NumPy*. O'Reilly Media, 2012.
- [15] Yotam Mann. The nsynth dataset, Abril 2017. [Online; NSynth Dataset].
- [16] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with wavenet autoencoders. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1068–1077. JMLR. org, 2017.
- [17] Raphael Campos. Machine Learning Beyond Deep Learning Árvores de decisão. <https://medium.com/machine-learning-beyond-deep-learning/%C3%A1rvores-de-decis%C3%A3o-3f52f6420b69>. Accessed: 2019-12-04.
- [18] Diego Lopez Yse. Towards data science: The complete guide to decision trees. Accessed: 2019-12-04.
- [19] Márcio Porto Basgalupp. *Árvores de Decisão*. PhD thesis, Universidade Estadual de Campinas - UNICAMP, 2010.
- [20] Gilbert Tanner. Towards data science: A guide to ensemble learning: Increase your accuracy by combining model outputs, 2019. Accessed: 2019-12-04.
- [21] G. E. P. BOX. NON-NORMALITY AND TESTS ON VARIANCES. *Biometrika*, 40(3-4):318–335, 12 1953.

Instance Selection for Music Genre Classification using Heterogeneous Networks

Angelo Cesar Mendes da Silva¹, Paulo Ricardo Viviurka do Carmo¹,
Ricardo Marcondes Marcacini¹, Diego Furtado Silva²

¹Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, Brazil
Av Trabalhador São-carlense, 400, Centro – 13566-590 São Carlos, SP

²Departamento de Computação – Universidade Federal de São Carlos, Brazil
Rod. Washington Luís, km 235, Jardim Guanabara – 13565-905 São Carlos, SP

{angelo.mendes, paulo.carmo, ricardo.marcacini}@usp.br, diegofs@ufscar.br

Abstract. In scenarios involving musical data, there are usually high-dimensional data and different modalities, such as audio and text, that cost more in machine learning tasks. Instance selection is a promising approach as pre-processing step to reduce these challenges. With the intent to explore the multimodality in music information, we introduce musical data instance selection into heterogeneous network models. We propose and evaluate ten different heterogeneous networks to identify more representative relationships with various musical features related, including songs, artists, genres, and melspectrogram. The results obtained allow us to define which network structure is more appropriate considering the volume of available data and the type of information that the features have. Finally, we analyze the relevance of the musical features, and the relationship does not contribute for instance selection.

1. Introduction

Data mining and machine learning methods for large volumes of data require expensive computational resources. Moreover, there are usually high-dimensional data and different modalities in scenarios involving musical data, such as audio and text. Instance selection is a promising approach to dealing with these challenges. Such methods aim to select a representative subset S from the complete dataset T , in which the performance function P of a machine learning method M is not significantly reduced, i.e., $P(M, S) \approx P(M, T)$. Although instance selection has been a research topic investigated for decades in the field of data mining and machine learning, most existing methods focus on unimodal scenarios, such as low-level characteristics of music audio signals.

Musical data can be represented by different data modalities such as lyrics, tags, and features extracted directly from audio content [1, 2]. Audio data is widely explored to extract features in different levels that represent the music in a vector space and provide input data for machine learning methods. For example, melspectrogram is used in genre classification [3, 4], and chromagram in the cover identification problem [5, 6]. More recently, deep learning methods have been used to learn representations of music from raw audio data, both for genre classification and instrument detection tasks [7, 8].

Multimodality in music information has been explored as complementary ways to build representations

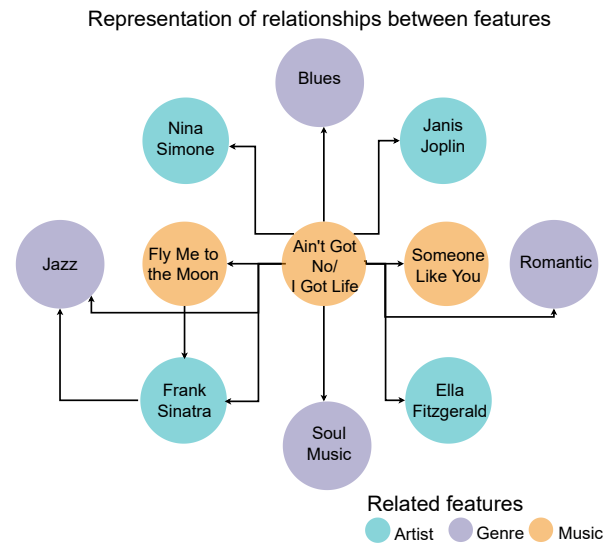


Figure 1: Possible relationships between artists, genres, and songs related to a song.

that have more discriminatory power [9, 10]. Exploring musical multimodality is a perceptive activity inherent to humans, so we can easily perceive relationships between different genres, similar artists, or songs that seem to belong to one artist. In general, these relationships are features that can be represented as metadata and are being highlighted as complementary musical information [11, 12].

A promising strategy to abstract and map these relationships is to build a model based on a heterogeneous network architecture for different modalities. Heterogeneous networks can relate the modalities following a specific similarity criterion that allows us to make associations between songs through its metadata, such as related genres, songs, and artists. Following this idea, we hypothesize that if we can build a model on the relationships between musical features, as illustrated in Figure 1, we can also use them to identify the most representative songs and use them as input for machine learning tasks.

In this paper, we introduce musical data instance selection from heterogeneous network models. We present a study on ten different heterogeneous networks to identify

more representative relationships from various modalities of the musical data. We evaluate the instance selection for musical genre classification, in which we aim to obtain a reduced subset to train a classification model. Our main contributions are threefold:

- We propose and evaluate heterogeneous networks with the most usual relationships in musical datasets, including songs, artists, genres, and mel-spectrogram features extracted from the audio signal. We discuss the relevance of the relationships between artists, songs, genres, and clusters to emphasize the features' particularities, and we show what kind of relationship does not contribute for instance selection.
- Due to the multimodality (audio and text) of the dataset, we investigate a regularization framework for heterogeneous networks. We explore regularization through label propagation, in which labeled vertices of the training set smoothly propagate their labels across the heterogeneous network. Thus, each vertex of the heterogeneous network has a membership vector with a label, which is then used to identify more representative instances.
- We present a score ranking function using the regularizer membership information obtained by the regularizer. Three different instance selection strategies based on ranking function were proposed and evaluated.

We carried out an experimental evaluation involving the 4MuLA (Multitask, Multimodal, and Multilingual Music Lyrics and Audio features) dataset [13]. The experimental results compare ten different heterogeneous networks and three instance selection strategies. In addition, two traditional methods of instance selection were used as reference models. The experimental results show evidence that heterogeneous networks are competitive for song instance selection in multimodal scenarios, especially with relationships involving cluster-based features from audio signals, as well as related genres and songs.

2. Related works

In this session, we explore existing approaches to the instance selection problem and when no have performance gains in problem-solving. We extend the discussion to discuss the characteristics of the methods and the specific needs for musical data problems. In addition, we discussed the elements of musical datasets and how using instance selection could mitigate their limitations.

In a classification problem, it is expected that data existent in the same class are contained in the same cluster and that there is a similarity between their features. However, there are scenarios where specific data have inconsistencies, and its structure does not match the other data in the group. To reach a good solution to a classification problem, strategies for selecting reduced subsets of instances and removing noisy instances have often been

explored. Although classification models are expected to perform better with a higher quality training set, it is also an interesting result to use instance selection to reduce the training set and maintain the classifier performance.

Among the possibilities for applying the instance selection process is isolating a reduced instance subset with a high representation of the data of the class information [14]. Similarly, we can also think that the process aims to remove superfluous or harmful samples that can be used to represent the class. We can understand this representation by observing data that have standard structural features about other data in your cluster, and an association between it and its class can be done objectively. The better strategy for selecting the most representative instances depends on an analysis of this structure [15].

Thus, the reduced set can be used as a representative of the class in several tasks and is expected to maintain or improve performance compared to the original set [16]. Furthermore, there is a wide variety of applications that have the selection process as a preliminary step [17], as an example, we can highlight noise identification [18], under-sampling in unbalanced classes [19], or missing data imputation [20].

Defining how to carry out the selection process is a task that requires computational cost evaluation, as there are scenarios where the selection cost is high and does not reflect a high-performance gain in the solution of a task [15, 17, 21]. Initial approaches proposed selection methods based on the analysis of closest instances [22, 23]. Even having a considerable cost, they are simple methods, and their concept is derived from other proposals that incorporate improvements [24, 25, 26]. Other works use models to classify instances to indicate the best candidates to represent the class, either through individual classifiers [27, 28] or through ensembles of models that have been shown to be efficient in identifying the particularities of the dataset [29, 30, 31].

In the context of musical data, problems with real-world scenarios naturally have unbalanced data regardless of the label type. For example, whether for commercial or cultural reasons, issues that have their data organized by genre, artists, or categorical tags, we are always seeing new music appear and be incorporated into categories without a pre-defined proportion [32]. In addition to these characteristics, music data is commonly represented by acoustic information and textual attributes such as lyrics and metadata that can express various information, for example, categories or relationships existing between a song and other songs, artists, or genres [33, 34]. Associating with human's musical perception, we can imagine that these features are complementary [35] and helpful to compose the most robust representation for a music dataset [36, 37].

The exploration of this type of problem using musical data still need further investigation. The granularity of metadata information is a known limitation that makes it difficult to use different data sources to represent songs [38]. However, this difficulty can be mitigated with

the availability of new datasets in recent years [39] and direct access to data via proprietary APIs^{1,2,3} and with possibilities to design models that explore the relationships between features to build a representation of this information. Models based on heterogeneous networks have this characteristic due to the possibility of incorporating different features such as nodes in their architecture, exploring the relationships between them, and propagating information that allows reaching data with different similarity evaluation criteria.

In this paper, we will explore the existing gap regarding heterogeneous network, music multimodality and instance selection methods. We explore heterogeneous networks to model different types of information e determine relationships among songs that share the same attributes in a way that allows us to measure the relevance of music for its class.

3. Music Instance Selection from Heterogeneous Networks

Our instance selection method is based on two steps. The first step involves the generation of heterogeneous networks from the multiple modalities of the music dataset. The second step consists of a method for heterogeneous network regularization to generate a ranking function for each song on the network.

The generation of the heterogeneous network is performed by processing the music training set. The network relationships are extracted by different music metadata, as illustrated in Table 1. The relationships between songs and their metadata are mainly generated through k-partition network structures. In Figure 2, we can observe a representation of the network #4, where the network has a 3-partite structure connected by a constructed feature using the k-means⁴ algorithm partitions from the melspectrogram data and the related artist features.

All cluster-based features are obtained using the k-means algorithm. The k parameter (number of clusters) was chosen by an iterative method that searched the highest silhouette⁵ measure considering all the k values between $k = 2$ and $k = (class\_size/2)$.

We used traditional features that can be easily obtained from music metadata, and raw audio signals. They are described as follows:

- **Song:** is the `music_id` that can be referenced to other features like: melspectrogram, metadata, lyrics, and others;
- **k-means cluster:** is the k-means cluster allocated for the song;

¹Last.fm API. Available at <https://www.last.fm/api>.

²Spotify API. Available at <https://developer.spotify.com/documentation/web-api/>.

³Pandora API. Available at <https://developer.pandora.com/>

⁴K-means is a clustering algorithm that partitions the data minimizing the mean squared error when selecting points near a synthetic center point called centroid [40]

⁵Silhouette is a validation measure for cluster partitions [41]

Table 1: Overview of the network construction strategy.

#Network	Features	#K
1	Song and k-means cluster	2
2	Song, k-means cluster and closest_cluster	3
3	Song, k-means cluster and sub_cluster	3
4	Song, k-means cluster and related_art	3
5	Song, k-means cluster and related_genre	3
6	Song, k-means cluster and related_music	3
7	Song, k-means cluster, related_art and related_genre	4
8	Song, k-means cluster, related_art and related_music	4
9	Song, k-means cluster, related_genre and related_music	4
10	Song, k-means cluster, related_art, related_music and related_genre	5

- **closest_cluster:** is the k-means cluster with the closest centroid to the song, apart from it's cluster;
- **sub_cluster:** is another k-means cluster that was executed within the genre of the song using the silhouette index to define ideal number of sub clusters;
- **related_art:** are all the related artists to a song according to dataset metadata.
- **related_genre:** are all the related genres to a song according to dataset metadata.
- **related_music:** are all related songs to a song according to dataset metadata.

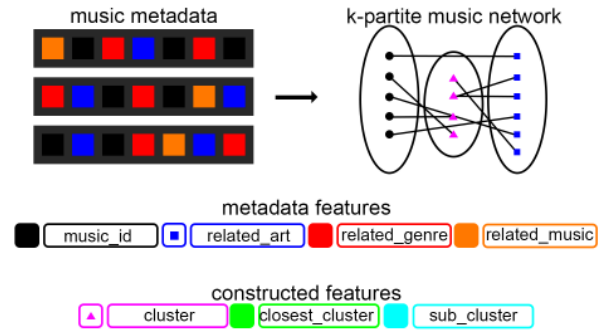


Figure 2: Example of a k-partite network used for feature extraction.

In the generation of heterogeneous networks, both the relationships extracted from cluster-based features and relationships extracted by metadata information on music genres, songs, and artists generate an unweighted network. More formally, we develop a heterogeneous network $N(O, R)$, where O is a set of nodes in the network and R is the relations between the nodes. We use an $O_L \subset O$ node subset (labeled song) for the second step of our method, i.e., the regularization framework defined in Equation 1.

$$Q(\mathbf{F}) = \frac{1}{2} \sum_{o_i, o_j \in O} r_{o_i, o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2 + \mu \sum_{o_i \in O_L} (\mathbf{f}_{o_i} - \mathbf{y}_{o_i})^2 \quad (1)$$

We want to find a matrix $\mathbf{F}$ that minimizes Equation 1. The node label information of the subset $o_i \in O_L$ is represented by the vector $\mathbf{y}_{o_i}$. A randomly initialized vector $\mathbf{f}_{o_i} \in \mathbf{F}$ is associated with each node o_i in the network.

The first term, $\frac{1}{2} \sum_{o_i, o_j \in O} r_{o_i, o_j} (\mathbf{f}_{o_i} - \mathbf{f}_{o_j})^2$, aims to minimize the distance between the label information of two neighboring vertices in the network according to the relation r_{o_i, o_j} . The second term, $\mu \sum_{o_i \in O_L} (\mathbf{f}_{o_i} - \mathbf{y}_{o_i})^2$, aims to minimize the distance between the vector $\mathbf{f}_{o_i}$ estimated for labeled vertices $o_i \in O_L$. The higher the μ parameter value, the greater the preservation of label information during regularization.

Our regularization framework is a specific case of the LLGC method (Learning with local and global consistency) [42]. We use a label propagation strategy to minimize Equation 1. The strategy is based on random walking, where for each iteration, a node propagates its label information to its neighbors, considering the probability of a path in the network. As a result, the process converges after some iterations, in which the $\mathbf{F}$ matrix shows little change between consecutive iterations.

The matrix $\mathbf{F}$ resulting from the regularization process can be interpreted as membership of the network nodes for each label. Our instance selection process exploits the difference between the membership vector $\mathbf{f}$ and the real label information $\mathbf{y}$ to determine a song's score, as defined in Equation 2. The lower this score, the greater the evidence that the song is well allocated in its original label considering all relationships in the network. On the other hand, the higher the score, the song represents more challenging instance of the classification problem, for example, allocated in the decision boundary between two or more classes.

$$score(o_i) = \|\mathbf{f}_{o_i} - \mathbf{y}_{o_i}\|^2 \quad (2)$$

For the instance selection, we set up three scenarios, that select only the amount we want for each training step, described as follows:

1. **Instance Selection L-Scores:** we select the m instances with the lowest score values.
2. **Instance Selection H-Score:** we select the m instances with the highest score values.
3. **Instance Selection LH-Score:** We selected m instances composed of the lowest and highest score values, i.e., a combination of the two previous strategies.

The first L-Score scenario aims at instance selection with "easier" instances for the classification model. However, such a strategy can generate classification models with very simple decision functions, which affect the classification of unseen data. On the other hand, the H-Score scenario aims at instance selection with "harder" instances that are more difficult to classify, generating more complex decision functions that are susceptible to overfitting. Thus, the LH-Score scenario is an attempt to obtain a trade-off between the two previous scenarios.

4. Experimental analysis

In this section, we describe the settings of the experiments carried out with the proposal evaluation scenarios. We also

discussed the network architectures built and the performance obtained in each scenario.

4.1. Experimental setup

To evaluate our proposal, we use two versions of the 4MuLA [13] music dataset: Tiny (T-4MuLA) and Small (S-4MuLA). The 4MuLA dataset contains melspectrogram information extracted from audio content and various metadata like genres, artists, and related songs. We have chosen it due to the larger musical diversity and the feature structure, like the relationships between artists, songs, and genres required in our network-building process. A summary is shown in Table 2.

Table 2: Overview of the datasets' characteristics.

	T-4MuLA	S-4MuLA
Songs	1988	9661
Artists	93	491
Genres	27	51
Min instances in genre	9	11
Max instances in genre	350	1169
Avg instances in genre	73.62	189.43
Std instances in genre	82.26	260.86

We experimented with all approaches in our proposal, selecting different percentages of training data in both datasets. Thus, we compare our results with specialist and random selection approaches, both using the same input training data. The specialist approach was constructed using a Naive Bayes classifier to indicate an instance's confidence score. The confidence was considered when the class predicted is equal to the true class. The train and test steps occurred using the Leave-One-Out cross-validation strategy to have the max instance by class tagged with the score. We highlight that we created the train and test subsets from the original dataset. However, we showed only the train data to the approach to select the more representative instances. The test subset was shown only in the evaluation step as unseen data. It is important to mention that all experiments were executed 10 times using different seeds to randomize the division of the data.

To obtain classification performance results, we trained a Random Forest classifier using the more representative instances indicated by all approaches in each data percentage for all scenarios aforementioned. We chose the F1-score as evaluation metric. The goal is to solve a multiclass classification to identify the genre of an unknown test split to measure the quality of selected instances, and to discuss the metadata relationship relevance to music representation. We discuss the experimental results considering two aspects: (1) an overview of the F1-score in each heterogeneous network compared to specialist and random approaches; and (2) a behavior analysis for the best scenario of each method with an increase in the training set.

4.2. Network structure

Figure 3 shows the results in 5 scenarios for the T-4Mula dataset. The boxplot contains the F1-score distribution

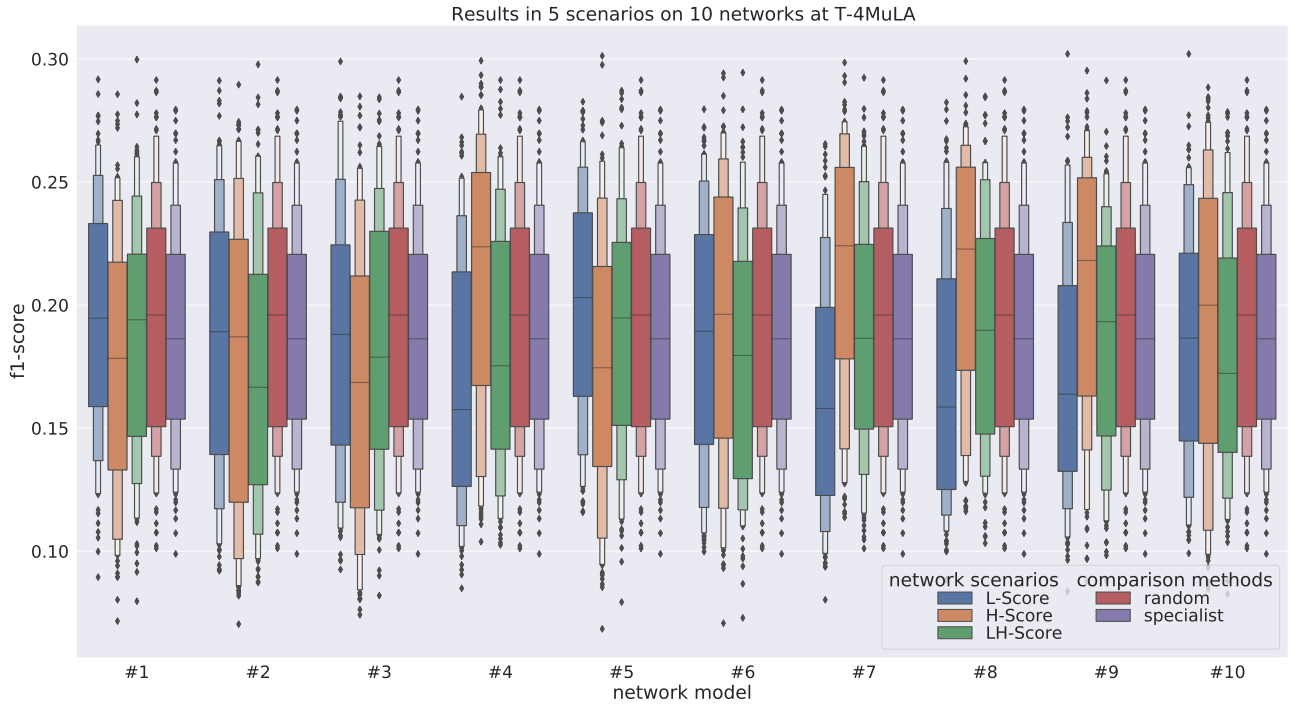


Figure 3: Results overview for the T-4MuLA dataset.

of all networks in the evaluated scenarios together with the two reference approaches. These results disregard the training set size. From this plot, we can observe that most of the methods achieve approximate performance.

The initial three network architectures have nodes formed by cluster information obtained only by melspectrogram. However, due to the difficulty of conceptualizing and assigning a musical genre considering only the audio content, we noticed that only this type of information has limitations highlighting more representative songs for a genre. In this way, randomly selecting instances proved to be more efficient.

In the following networks, we have the metadata relationship information, and we can observe a contribution in the discrimination of the songs. The using the more connected instances to training got better performance, with much higher values the median of comparable methods. It is essential to mention that four networks #4, #7, #8, and #10 have connections with nodes representing the related artists of the songs, indicating that this feature contains helpful information.

The results we obtained for the S-4MuLA dataset differ from T-4MuLA in some key areas. For example, in Figure 4 we can see that the L-Score scenario obtained better results than others scenarios in most of the networks. However, the random sampling got overall better results or was similar to the network approach.

Unlike classification results from T-4MuLA, networks #5, #6, and #10 obtains better classification performance. When analyzing the attributes they have in common, we notice that the relationships between similar genres and similar music are present. These relationships al-

low us to infer that for a scenario with a large volume of music, the information that indicates the relationship between artists and songs is relevant, and songs and related genres tend to be useful for classification.

4.3. Best scenarios

To provide another point of view from the results, we selected the best scenarios within our general plot to have a deeper look at how each feature changes the performance within the network. In the tables 3 and 4 we show F1-score behavior with an increasing training set. We selected the iteration in which the approach had the highest mean F1-score, and we established the individual value for each percentage.

Both tables show us that when considering a scenario where we have relationships capable of helping in musical discrimination, the instance selection process proves to be valid as a pre-processing step for the genre classification task. With an increase in the volume of training data, the performance of the proposed network increased the difference with the comparison methods. Such results show that instance selection from heterogeneous networks is competitive, but it requires careful tuning of the regularization process and the choice of scenarios. This strategy can be explored with visualization tools, taking advantage of interpretability and visual data exploration in heterogeneous networks.

We performed Friedman’s statistical test with Nemenyi’s post-test [43] to compare the scenarios to selection instances and the networks proposed. Figure 5 presents the result of the Friedman test with Nemenyi’s post-test through the critical difference diagram. The approaches

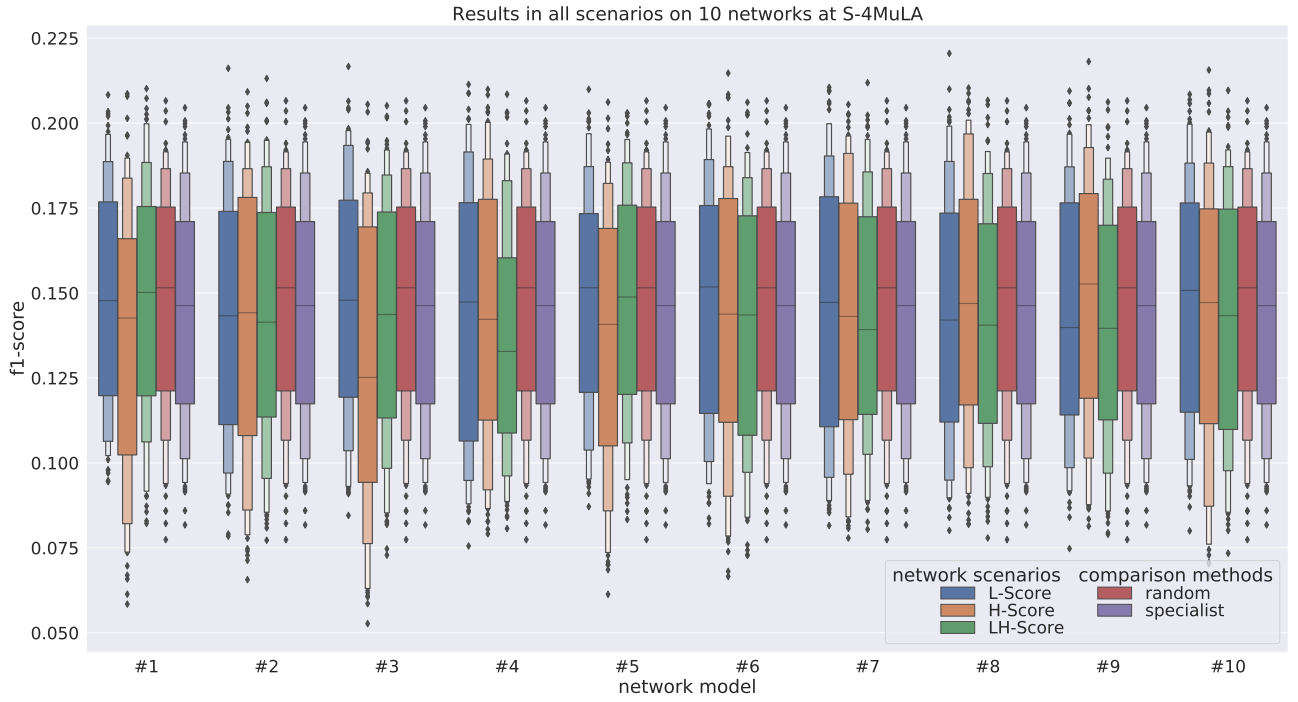


Figure 4: Results overview for the S-4MuLA dataset.

Table 3: Results considering the training size increase in the T-4MuLA dataset

F1-score										
Approach	Percentage of training set for instance selection.									Scenario
	10%	20%	30%	40%	50%	60%	70%	80%	90%	
Network #1	0.0802	0.0983	0.1254	0.1552	0.1737	0.2152	0.2425	0.2461	0.2857	L-Score
Network #2	0.1087	0.1171	0.1224	0.1518	0.2012	0.2007	0.2233	0.2492	0.2642	L-Score
Network #3	0.0807	0.0830	0.1090	0.1456	0.1779	0.2090	0.2398	0.2537	0.2848	L-Score
Network #4	0.1147	0.1624	0.1990	0.2237	0.2472	0.2651	0.2934	0.2853	0.2993	H-Score
Network #5	0.0684	0.1067	0.1109	0.1385	0.1745	0.1825	0.1953	0.2474	0.2547	L-Score
Network #6	0.1125	0.1597	0.1687	0.1761	0.2236	0.2372	0.2694	0.2850	0.2727	H-Score
Network #7	0.1290	0.1519	0.1965	0.2493	0.2662	0.2708	0.2814	0.2985	0.2931	H-Score
Network #8	0.1183	0.1561	0.2082	0.2376	0.2541	0.2695	0.2810	0.2920	0.2991	H-Score
Network #9	0.1686	0.1630	0.2170	0.2381	0.2393	0.2337	0.2555	0.2536	0.2724	H-Score
Network #10	0.1150	0.1398	0.2000	0.2045	0.2166	0.2306	0.2418	0.2802	0.2771	H-Score
specialist	0.1568	0.1492	0.1537	0.1823	0.2131	0.2385	0.2545	0.2327	0.2749	-
random	0.1419	0.1407	0.1913	0.1960	0.2135	0.2395	0.2430	0.2710	0.2616	-

Table 4: Results considering the training size increase in the S-4MuLA dataset

F1-score										
Approach	Percentage of training set for instance selection.									Scenario
	10%	20%	30%	40%	50%	60%	70%	80%	90%	
Network #1	0.0937	0.1155	0.1341	0.1453	0.1569	0.1719	0.1820	0.1964	0.2043	LH-Score
Network #2	0.0997	0.1082	0.1384	0.1554	0.1653	0.1662	0.1817	0.1937	0.2161	L-Score
Network #3	0.0971	0.1139	0.1355	0.1431	0.1728	0.1807	0.1928	0.1984	0.2166	L-Score
Network #4	0.0867	0.1034	0.1212	0.1349	0.1540	0.1718	0.1952	0.2005	0.2083	H-Score
Network #5	0.0945	0.1236	0.1322	0.1478	0.1510	0.1690	0.1954	0.2012	0.2099	L-Score
Network #6	0.1013	0.1137	0.1291	0.1437	0.1569	0.1694	0.1841	0.1974	0.2056	L-Score
Network #7	0.0971	0.0954	0.1181	0.1471	0.1607	0.1679	0.1906	0.1966	0.2105	L-Score
Network #8	0.0900	0.1068	0.1240	0.1458	0.1480	0.1837	0.1896	0.2028	0.2087	H-Score
Network #9	0.0871	0.1057	0.1237	0.1461	0.1658	0.1781	0.1980	0.2076	0.2181	H-Score
Network #10	0.1039	0.1108	0.1286	0.1551	0.1569	0.1753	0.1887	0.1938	0.2057	L-Score
specialist	0.0973	0.1045	0.1254	0.1374	0.1413	0.1647	0.1710	0.1909	0.1999	-
random	0.0951	0.1105	0.1234	0.1343	0.1565	0.1643	0.1740	0.1868	0.1921	-

connected by a line do not have statistically significant differences between them. For the scenarios, as seen in figures 3 and 4, the performance of networks, regardless of

the scenario, has similar values. According to this diagram, we cannot point to any superiority on a statistical basis. However, alone, we can note that the LH-Score and

random scenarios have a specific difference. We can relate this difference with the definition of both scenarios, where the first is formed by a combination of instances more significant with potential for increase the discrimination, while the second is a random selection of instances. The other scenarios show a more considerable similarity between the decision functions, regardless of the instance selection strategy.

About networks, there is also no statistically superior architecture. However, we can analyze the contribution of the features used by comparing the structure of the networks. For example, we can discuss the relevance of nodes with information related to the closest cluster and subcluster existing only in networks #2 and #3, as well as evaluate the relationships between the metadata of songs, artists, and genres used as an individual or combined feature in the other networks. Due to the similarity of the results, the possibility of discarding or prioritizing a specific type of node enables a cost reduction in the network construction process. Furthermore, it tends to impact in the discrimination of instances.

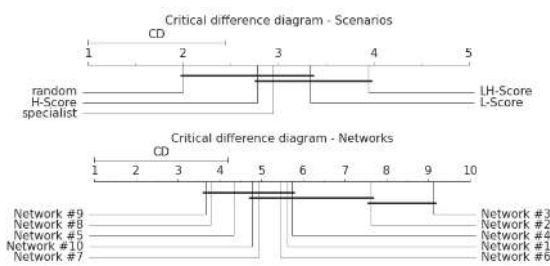


Figure 5: Analysis of the critical difference between scenarios and networks.

5. Conclusion

In this work, we present an evaluative study on the instance selection task using multimodal musical data. We built ten different heterogeneous network architectures that contained information based on melspectrogram and metadata to handle dataset multimodality, indicating the relationship between a song and attributes such as artist, genres, and related pieces. We compare our proposed approach to the (i) method with information from a specialist model to indicate the ideal instances for training and (ii) a random sampling selection process.

In general, the random sampling process for selecting instances proved to be efficient and less costly. However, by analyzing each approach individually, we observed the relevance of metadata and their relationships to compose a musical identification. From this observation, we can identify conditions where there is no large volume of data available and assess what kind of information is helpful knowledge for instance selection.

In future work, we intend to develop an application of this proposal as a web service where users can propose nodes and relationships to qualify the music dataset and create personalized scenarios where networks can con-

tribute to identifying unique characteristics of the musical genres.

6. Acknowledgments

This work was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code PROEX-12049601/D and 88887.513429/2020-00, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – Finance Code 2019/25010-5 and 2019/07665-4, Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPQ) – Finance Code 426663/2018-7.

References

- [1] Markus Schedl, Emilia Gómez, Julián Urbano, et al. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2-3):127–261, 2014.
- [2] YV Murthy and Shashidhar G Koolagudi. Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. *ACM Computing Surveys*, 51(3):45, 2018.
- [3] Keunwoo Choi, George Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2392–2396, 2016.
- [4] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.
- [5] Joan Serrà, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. Springer, 2010.
- [6] Diego F. Silva, Chin-Chia M. Yeh, Gustavo E. A. P. A. Batista, and Eamonn Keogh. SiMPle: assessing music similarity using subsequences joins. In *International Society for Music Information Retrieval Conference*, pages 23–29, 2016.
- [7] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmman, and Xavier Serra. End-to-end learning for music audio tagging at scale. In *International Society for Music Information Retrieval Conference*, pages 637–644, 09 2018.
- [8] Christine Dewi, Rung-Ching Chen, Yan-Ting Liu, et al. Similar music instrument detection via deep convolution yolo-generative adversarial network. In *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*, pages 1–6. IEEE, 2019.
- [9] W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.
- [10] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 14(3):478–493, Mar 2020.
- [11] Y. Lin, C. Chung, and H. H. Chen. Playlist-based tag propagation for improving music auto-tagging. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2270–2274, 2018.

- [12] Y. Lin and H. Chen. Tag propagation and cost-sensitive learning for music auto-tagging. *IEEE Transactions on Multimedia*, pages 1–1, 2020.
- [13] Angelo Cesar Mendes da Silva, Diego Furtado Silva, and Ricardo Marcondes Marcacini. 4mula: A multitask, multimodal, and multilingual dataset of music lyrics and audio features. In *Proceedings of the Brazilian Symposium on Multimedia and the Web, WebMedia '20*, page 145–148, New York, NY, USA, 2020. Association for Computing Machinery.
- [14] J. Arturo Olvera-López, J. Ariel Carrasco-Ochoa, J. Francisco Martínez-Trinidad, and Josef Kittler. A review of instance selection methods. *Artificial Intelligence Review*, 34:133–143, 2010.
- [15] Mellish C Brighton H. Advances in instance selection for instance-based learning algorithms. *Data Mining and Knowledge Discovery*, 6:153–172, 2002.
- [16] van der Aalst Wil Sani Mohammadreza Fani, van Zelst Sebastiaan J. Improving the performance of process discovery algorithms by instance selection. *Computer Science and Information Systems*, 17:927–958, 2020.
- [17] Marek Grochowski and Norbert Jankowski. Comparison of instance selection algorithms ii. results and comments. In Leszek Rutkowski, Jörg H. Siekmann, Ryszard Tadeusiewicz, and Lotfi A. Zadeh, editors, *Artificial Intelligence and Soft Computing - ICAISC 2004*, pages 580–585, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [18] Jianping Zhang, Yee-Sat Yim, and Junming Yang. *Intelligent Selection of Instances for Prediction Functions in Lazy Learning Algorithms*, pages 175–191. Springer Netherlands, Dordrecht, 1997.
- [19] Chih-Fong Tsai, Wei-Chao Lin, Ya-Han Hu, and Guan-Ting Yao. Under-sampling class imbalanced datasets by combining clustering analysis and instance selection. *Information Sciences*, 477:47–54, 2019.
- [20] Chih-Fong Tsai and Fu-Yu Chang. Combining instance selection for better missing value imputation. *Journal of Systems and Software*, 122:63–71, 2016.
- [21] Joel Luís Carbonera and Mara Abel. Efficient instance selection based on spatial abstraction. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 286–292, 2018.
- [22] Peter Hart. The condensed nearest neighbor rule. *IEEE transactions on information theory*, 14(3):515–516, 1968.
- [23] G Ritter, H Woodruff, S Lowry, and T Isenhour. An algorithm for a selective nearest neighbor decision rule. *IEEE Transactions on Information Theory*, 21(6):665–669, 1975.
- [24] Joel Luis Carbonera and Mara Abel. A density-based approach for instance selection. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 768–774, 2015.
- [25] B. Ramesh and J.G.R. Sathiaselvan. An advanced multi class instance selection based support vector machine for text classification. *Procedia Computer Science*, 57:1124–1130, 2015. 3rd International Conference on Recent Trends in Computing 2015 (ICRTC-2015).
- [26] Junhai Zhai, Xizhao Wang, and Xiaohe Pang. Voting-based instance selection from large data sets with mapreduce and random weight networks. *Information Sciences*, 367-368:1066–1077, 2016.
- [27] Miguel Lopes, Fabien Gouyon, Alessandro L. Koerich, and Luiz E.S. Oliveira. Selection of training instances for music genre classification. In *2010 20th International Conference on Pattern Recognition*, pages 4569–4572, 2010.
- [28] Ismail M. Anwar, Khalid M. Salama, and Ashraf M. Abdelbar. Instance selection with ant colony optimization. *Procedia Computer Science*, 53:248–256, 2015. INNS Conference on Big Data 2015 Program San Francisco, CA, USA 8-10 August 2015.
- [29] Nicolás García-Pedrajas and Aida de Haro-García. Boosting instance selection algorithms. *Knowledge-Based Systems*, 67:342–360, 2014.
- [30] Aida de Haro-García, Gonzalo Cerruela-García, and Nicolás García-Pedrajas. Instance selection based on boosting for instance-based learners. *Pattern Recognition*, 96:106959, 2019.
- [31] Marcin Blachnik. Ensembles of instance selection methods: A comparative study. *International Journal of Applied Mathematics and Computer Science*, 29(1):151–168, 2019.
- [32] Wenqin Chen, Jessica Keast, Jordan Moody, Corinne Moriarty, Felicia Villalobos, Virtue Winter, Xueqi Zhang, Xu-anqi Lyu, Elizabeth Freeman, Jessie Wang, Sherry Kai, and Katherine M. Kinnaird. Data usage in mir: History & future recommendations. In *International Society for Music Information Retrieval Conference*, 2019.
- [33] Sergio Oramas, F. Barbieri, Oriol Nieto, and Xavier Serra. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval*, 1:4–21, 2018.
- [34] Ngo Tung Son, Duong Xuan Hoa, and Vu Thanh. Learning sparse representation from multiple-source data for relative similarity in music. In *Proceedings of the 2018 International Conference on Computational Intelligence and Intelligent Systems*, CIIS 2018, page 1–4, New York, NY, USA, 2018. Association for Computing Machinery.
- [35] Ho-Hsiang Wu, Chieh-Chi Kao, Qingming Tang, Ming Sun, Brian McFee, Juan Bello, and Chao Wang. Multi-task self-supervised pre-training for music classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 556–560, 06 2021.
- [36] Y.R. Pandeya and J Lee. Deep learning-based late fusion of multimodal information for emotion classification of music video. *Multimed Tools Appl*, 2020.
- [37] Changfeng Chen and Qiang Li. A multimodal music emotion classification method based on multifeature combined network classifier. *Mathematical Problems in Engineering*, 2020.
- [38] Michael I. Mandel and D. Ellis. Multiple-instance learning for music information retrieval. In *International Society for Music Information Retrieval*, pages 577–582, 2008.
- [39] Rachel M Bittner, Magdalena Fuentes, David Rubinstein, Andreas Jansson, Keunwoo Choi, and Thor Kell. mir-data: Software for reproducible usage of datasets. In *International Society for Music Information Retrieval (ISMIR) Conference*, 2019.
- [40] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.
- [41] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.
- [42] Dengyong Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004.
- [43] Bogdan Trawinski, Magdalena Smetek, Zbigniew Telec, and Tadeusz Lasota. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. *Applied Mathematics and Computer Science*, 22(4):867–881, 2012.

Tempo estimation via neural networks – a comparative analysis

Mila Soares de Oliveira de Souza¹, Pedro Nuno de Souza Moura¹, Jean-Pierre Briot^{1 2}

¹ Escola de Informática Aplicada – Universidade Federal do Estado do Rio de Janeiro
Av. Pasteur, 458 – 22290-255 Rio de Janeiro, RJ

² Sorbonne Université, CNRS, LIP6, F-75005 Paris, France

milasoaresdeoliveira@gmail.com, pedro.moura@uniriote.br, Jean-Pierre.Briot@lip6.fr

Abstract

This paper presents a comparative analysis on two artificial neural networks (with different architectures) for the task of tempo estimation. For this purpose, it also proposes the modeling, training and evaluation of a B-RNN (Bidirectional Recurrent Neural Network) model capable of estimating tempo in bpm (beats per minutes) of musical pieces, without using external auxiliary modules. An extensive database (12,333 pieces in total) was curated to conduct a quantitative and qualitative analysis over the experiment. Percussion-only tracks were also included in the dataset. The performance of the B-RNN is compared to that of state-of-the-art models. For further comparison, a state-of-the-art CNN was also retrained with the same datasets used for the B-RNN training. Evaluation results for each model and datasets are presented and discussed, as well as observations and ideas for future research. Tempo estimation was more accurate for the percussion-only dataset, suggesting that the estimation can be more accurate for percussion-only tracks, although further experiments (with more of such datasets) should be made to gather stronger evidence.

1. Introduction

Music Information Retrieval (MIR) is the field of study that aims to extract, analyze and provide information from a song [1]. Basic methodologies for Music Information Retrieval can include audio signal processing, musical perception, among others [2]. Some examples of MIR research involve tempo estimation [3] — which is the main research theme of this paper —, identifying musical styles [4] and comparison of similarity between two songs [5]. Estimating the tempo of a song is considered one of the most fundamental tasks of MIR [6].

The tempo of a musical piece corresponds to the number of beats that would be counted in one minute (beats per minute, or bpm) [7]. The perception of tempo can also be a subjective phenomenon; when listening to a song, it is often generalized that music sounding “fast” to a listener should have a faster tempo than songs that seem comparatively slower [8].

Rhythm is one of the most important characteristics of music [7], and it can be said that it is one of the factors that makes certain musical genres easily recognizable

(e.g.: waltz in 3/4 and rock in 4/4-time signatures). Some of them more often work with faster tempo – like heavy metal – while others often work with comparatively shorter tempo – like reggae. Identifying such tempo could have interesting applications in other contexts, such as, for example, a recommendation system for similar songs, which would suggest tracks considering similar tempi and/or other characteristics.

As such, it could be interesting for research methods to estimate the tempo of a musical piece using modern technologies with good accuracy, as well as verifying if there is any model in such technology that stands out in performance. Still considering the importance of tempo, it would be also interesting to test whether the tempo estimation is more accurate when estimating over tracks containing percussion only, which is often one of the most responsible instrument(s) for determining tempo.

The tempo estimation in music using computational methods has been studied for some decades. As an example, the work of Schreier [9], from 1998, was one of the first to process the audio signal in a continuous way (previously, the usual methods worked over discrete temporal events) [10].

As for tempo estimation using artificial neural networks (ANNs), the first evidences were found at late 2000s-early 2010s. Some of these works were those of Böck [6], circa 2010, and that of Gkiokas, Katsouros and Carayannis [11] in 2012. Böck uses a bidirectional Long-Short Term Memory (LSTM) [12] in order to map beat activations, after which the results are further processed with comb filters. The work of [11] also uses a convolutional neural network to derive a beat activation function, then identify the tempo in later steps.

A neural network model considered state of the art (for tempo estimation) was proposed by Böck and Schedl in 2011 [13] and further refined in 2015 [10]. The model is a bidirectional LSTM, aided by external modules which process detected beat activations. The first published model capable of estimating tempo without the help of external modules that presented state of the art performance was the work of Schreiber and Müller [14] published in 2018, consisting of convolutional and dense layers. This model was an important starting point for the development of this work. The model on [14] presented results comparable to the state of the art. Böck and Davies,

in 2020 [15], proposed a refinement of a system (originally proposed by them in 2019 [16]), which uses a CNN with convolutional layers and TCN layers (temporal convolutional networks), then processes the results with deep Bayesian networks, also achieving state-of-the-art results. These studies suggest that developing neural networks can lead to the improvement of the estimation performance compared to more traditional algorithms.

Besides the use of datasets containing multiple musical instruments, we propose the use of datasets consisting of percussion instruments only (such as drums) to the estimation of tempo via neural networks research. One of the reasons this is yet to be more explored might be due to the lack of extensive availability of such databases to the MIR community. However, we found there are percussion-only databases being constructed, which already provide a reasonably good number of tracks and annotations to work with. As such, we decided to explore this possibility.

This paper also aims to propose, train and evaluate a new bidirectional recurrent neural network model (B-RNN) that is capable of estimating the tempo in bpm of a musical piece without the need for help of other additional modules (external to the model itself). The purpose of selecting this kind of model is to enrich the comparative analysis on different types of ANNs for tempo estimation. The ANNs should be capable of estimating tempo of any musical piece containing well-defined global temporal structure. We evaluate the performance over a collection of datasets, then compare and discuss the results considering other architectures such as convolutional neural networks.

2. Dataset

Name	Qt. of sample	Duration (s)	Extension	Genres
ACM [17]	1.410	30	.wav	Pop, rock
Extended Ballroom [18]	3.826	30	.mp3	Salsa, foxtrot, samba etc.
GiantSteps Tempo [19]	664	120	.mp3	Electronic Dance Music (EDM)
GiantSteps MTG ¹	1.158	120	.mp3	EDM
Groove ²	443	10-60+	.wav	Reggae, pop, rock, jazz etc.
GTZan [20]	999	30	.wav	Pop, rock etc.
Hainsworth [5]	222	40-60+	.wav	Folk, jazz etc.
LMD [21]	3.611	30	.mp3	Pop, rock, classical etc.

Table 1: Selected datasets and their features.

¹ Available on: <<https://github.com/GiantSteps/giantsteps-tempo-dataset>>. Access in: 30 apr 2021.

² Available on: <<https://ai.google/research/teams/brain/magenta/>>. Access in: 30 apr 2021.

For the experiment, we built a large dataset containing complete musical pieces (or samples). These pieces might contain either multiple instruments or a single instrument, such one drum line (which is a percussion instrument) without other instruments.

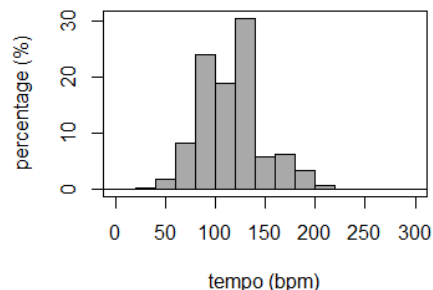


Figure 1: Tempo distribution of the dataset samples.

The dataset for this experiment contains 12,333 parts/samples. It is a collection of smaller datasets that were selected for their variety of pieces (diversity of musical genres, also avoiding repetition of data). The free availability for use in academic experiments — as well as availability of public annotations (of tempo in bpm) — in their respective papers/source were also important factors.

Datasets containing percussion-only tracks were also considered necessary for this experiment. This is due to the fact that percussion instruments, such as drums, are often the most used instrument to guide the rhythm of a song (particularly in pop and rock); thus, it is of our interest to verify whether the tempo estimation could be more accurate if the network tries to estimate the tempo only over the percussion. Furthermore, to the best of our knowledge there are no works on tempo estimation (with neural networks) that presented experiments with drum-only tracks (nor the Groove dataset). Should the network performance over drums-only pieces eventually be considerably more accurate, it could be interesting to try extracting the drum lines from a multi-instrument song for future experiments.

Table 1 lists the names, number of pieces and presents a brief description of each database, showing also the musical genres of the pieces that make up each one. After the name of each dataset, there is also a reference to their original publication, which grants the access to the musical pieces and the available annotations (exceptions: the annotations for GiantSteps MTG and LMD were obtained from [14] instead of their original sources).

The Groove dataset is the only that contains purely percussion pieces (in this case, drums), totaling 443 samples. All the others present mostly multi-instrument tracks, some of them containing vocal lines. This can be

considered an obstacle for the goal of determining whether there is greater accuracy in estimating the tempo of percussion-only pieces, since the amount of data is a crucial factor for the performance of a neural network [22]. Although the original Groove dataset on the source page is more extensive, most of the samples which are available in the Groove dataset consist of information that is not relevant to this experiment, such as drum fills. The authors filtered the database and annotations so that only tracks with proper instrumentation (and available .wav file) remained.

It is interesting to highlight a few points on the selected datasets: Extended Ballroom contains music from 13 different genres, as well as time signatures other than 4/4 (which is the most common in pop songs), such as 3/4. Groove also features popular drum lines in a variety of genres such as punk, rock, jazz, and gospel, also presenting a variety of time signatures including 3/4 and 6/8. It is possible to consider that Groove does not suffer from style bias, as the tracks were performed by several drummers. The diversity of styles and metrics is an enriching factor for the database.

3. Methodology

Tempo estimation of a musical piece (which is originally an audio file) can be approached as both a regression problem and a classification problem. The classification approach can consider the tempo in bpm of a song as a class; a class is an integer number which is between the lowest and the highest bpm values selected. This method could be interesting due to the possibility of working later with other tempo probabilities, such as finding a second or third tempo considered more suitable [14]. This information could also be relevant when analyzing a musical piece structured with multiple tempi (although this paper proposes to work with the scope of a single tempo only). Furthermore, decimal values can be deemed irrelevant for tempo estimation. For these reasons, the authors chose to work with the classification problem approach instead of regression.

After defining that the experiment will be conducted on neural networks – due to the state-of-the-art performance evidences [25] [15] –, the first step was the dataset selection (introduced in Section 2). Then the following steps consisted of defining the input representation for the neural network, defining the neural network model, network training and evaluation; these first three following steps are presented and discussed in this Section, while the evaluation of models and results are discussed in Section 4.

In short, we obtain a mel spectrogram of the music piece chosen as input, scale it on the time axis, crop it into several windows. These windows are the input for the neural network. The model will then process these windows, estimating the probabilities of the tempo classes for the windows; Finally, a single value of tempo in bpm will be chosen as output after considering all the windows probabilities. These processes and definitions will be further explained in this Section. It is expected that the model will identify global temporal structures in a

spectrogram.

It is worth noting that music does not necessarily consist of a single tempo in bpm; there are pieces that present bpm changes over time. Although the neural networks in this paper can detect and output multiple tempi that might constitute a musical piece – since it estimates the probabilities of different tempi in the entire piece –, we decided to work with the estimation of a single global tempo, meaning the final output is a single class of tempo in bpm. This is not only to simplify the approach to the problem, but mostly to ensure the possibility of performance comparison to the original datasets' annotations (which provide a single tempo value), as well as to other existing neural networks. Therefore, it is to be expected that the estimation of a global tempo will not be as accurate as possible for pieces without a major global tempo structure (due to the nature of the output). Future experiments could be done to explore the possibilities of using multiple tempi.

3.1 Input representation

For this experiment, the mel spectrogram (which will be defined in the next paragraphs) was selected as the input representation for the neural network. The input treatment process that will be described is replicated from [14].

Although the most basic form of signal representation is the waveform, which represents the magnitude (y-axis) over time, one of its downsides is being quite heavier than other representations for a system to process. Therefore, it was on the interest of the authors using Fourier transforms to obtain a spectrogram, which works in the context of frequency.

The spectrogram provides a two-dimensional visual representation of an audio signal, in which the horizontal axis represents time and the vertical axis represents frequency [23], while color schemes can be used to represent intensity (third axis). Figure 2 is an example of a spectrogram.

The mel spectrogram is a spectrogram that uses the Mel scale which represents frequencies akin to human perception [24]. The Mel scale is widely used in MIR research. Since tempo is defined according to human hearing, the Mel scale is useful to carry out experiments that aim to simulate hearing in a closer way to that of its creators.

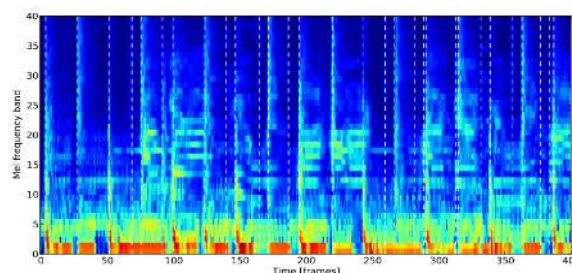


Figure 2: A mel spectrogram (extracted from [6]).

It is possible to process the input further for network training efficiency by scaling the spectrogram. The entire musical piece spectrogram can be scaled on the time axis (keeping the frequency axis untouched). For this experiment, the spectrogram is scaled along the time axis

using a random factor $c \in \{0.8, 0.84, \dots, 1.16, 1.2\}$ (the

tempo ground truth label is also scaled accordingly to factor c). This also generates diversity of inputs for further training of the network.

Instead of providing an entire spectrogram for the neural network to process, it is possible to crop several windows of this same spectrogram, which was the selected input method in this work. This mode has not only an advantage from a size efficiency point of view (input gets much smaller), but also from the number of examples for training. Cropping the spectrogram in windows, it is possible to generate several different training examples for the network from what was, originally, only one input.

The duration of 10 seconds can be considered enough to get a good sense of the local tempo of a part of a piece. Therefore, it is enough to choose a number of frames that is sufficiently close to such time interval. The value of 256 frames (approximately 11.9 seconds) for the spectrogram input, used by [25], was considered adequate.

Thus, the input to the neural network is a mel spectrogram with dimensions $F_T \times T_T = 40 \times 256$ (F_T is the number of *frequency bins* and T_T is the number of time frames). F_T covers the frequency range between 20 and 5,000 Hz.

There are also some interesting variations of the Mel spectrogram, such as the log Mel (which uses a logarithmic scale for intensity, that is, decibels). It may be interesting, for a future experiment (not in the scope of this paper), to use the log Mel spectrogram pursuing improvements in tempo estimation.

3.2 Model

For this experiment, we proposed a bi-directional recurrent neural network model as a preliminary step to assess the performance of different architectures of artificial neural networks.

Bidirectional recurrent neural networks (B-RNN), introduced by Schuster and Paliwal [26], are recognized as a good model for tasks such as speech recognition [28], also proving to be valuable for tempo estimation. Böck and Schedl [13] propose a B-LSTM (bidirectional Long Short-Term Memory, which is a type of B-RNN [28]) for tempo estimation; their model can provide state of the art accuracy [14], although they use external modules to process beat activation functions, which is not the goal of this paper.

The use of bidirectionality makes sense for identifying tempo patterns, since, when analyzing a song, both the previous context and the future context of a

moment in a song can be used to help determine the tempo value. Considering the aforementioned factors, the bidirectional recurrent neural network was chosen as a suitable type of model to be worked with in this paper. B-LSTMs could be an option as well, but the authors opted for a simpler architecture for the scope of this work. Future works could explore the use of B-LSTMs as a possibility of improvement.

Since the goal is to estimate tempo without the need of auxiliary modules external to the neural network itself, and the authors could not find any B-RNN models yet published that perform this task, a new model was proposed. The overview on the model is represented on Figure 3.

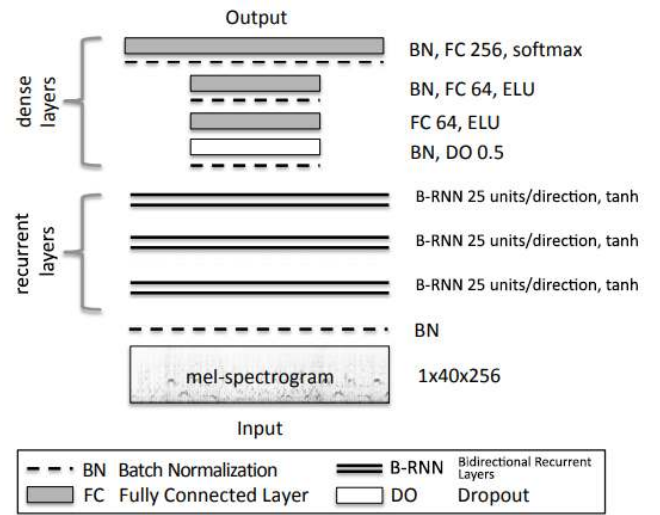


Figure 3: The B-RNN model overview (modified from [14]).

We constructed 3 B-RNN layers with 25 simple recurrent units for each direction to process the input. This corresponds to 6 layers and 150 units. The number of units was inspired on Böck and Schedl [13] B-LSTM. Since the B-RNN is simpler than the B-LSTM, it might be interesting to try more layers or units in the future (which goes beyond the scope of this paper). The recurrent layers have tanh (hyperbolic tangent) as the activation function. The input is normalized before it is sent to the recurrent layers.

The goal for recurrent layers is to identify onsets, analyzing the mel spectrogram frequencies, and identifying its temporal dependencies. Onset is considered the exact moment of the beginning of a beat; detecting them is needed to find its periodicities. With the bidirectional recurrency, we expect to detect sufficiently long temporal dependencies.

Since we do not want to use external modules for signal processing (such as comb filters), there is the need for the model to be able to process the output from the recurrent layers. Schreiber and Müller [14] proposed a CNN that use dense layers for such feature. Since their model also presented state-of-the-art results, we decided to replicate the dense layers from [14]. The replicated dense

layers can also be seen in Figure 3.

Thus, the next step is the processing by these dense layers, whose purpose is to classify the features detected by the recurrent layers. First, the output of the recurrent layers undergoes an average pooling of (5×1) . After a batch normalization as the next step, a Dropout layer ($p = 0.5$) is added to avoid overfitting, succeeded by a dense layer. A batch normalization followed by dense layers will then finish processing the data. The first two dense layers use ELU (exponential linear unit) [27] as the activation function, while the last one uses softmax.

Since the network deals with a multiclass classification problem, we chose categorical cross entropy for the error function. The chosen optimizer is SGD (stochastic gradient descent) with a clipping value of 5 to prevent gradient explosion. The value for learning rate is 0.001; for momentum it is 0.9. The bidirectional recurrent neural network has a total of 6,583,772 trainable parameters.

3.3. Training

For our experiment, it is necessary to select a part of the dataset that will be used to train the neural network. Thus, we must divide it in two distinct subsets: one for the training step, and another for the evaluation step later.

In order to avoid biased results, it was decided that datasets in the training subset cannot be in the test subset (and vice versa), with the exception of Groove (because it is the only one that contains drum-only tracks). It is desirable that the training subset contains a good variety of genres.

Based on these criteria, we selected for training the models the following subsets: Extended Ballroom, GiantSteps MTG, Hainsworth, LMD and part of Groove (90%), totaling 9,215 samples.

Among these datasets selected for training, it is necessary to split them into two more subsets, one which will be used for the network training itself and the other one which should be used to measure the validation loss. The 80% rate was set for the model training, with then 20% for validation. The tracks for each subset were selected randomly.

The only exception is Groove, as mentioned: since it does not contain an overwhelming number of samples (443 pieces only), also being the only dataset with percussion lines, we must consider a subset for evaluation. Groove was split following the proportions: 80% training, 10% validation and 10% evaluation. We made an effort to ensure that tracks used in evaluation were as different as possible from those in training and validation.

As discussed in Section 3.1, the Mel spectrogram of the entire signal is compressed (or expanded, if the duration of the entire audio is less than 11.9 s) and then cropped into smaller excerpts of approximately 11.9 seconds. This process therefore increases the number of inputs that are provided to the network during training.

The training was carried out under early stopping (when there is no validation loss in the last 100 Epochs).

The code used to run the training stage was originally made available in [25], with some minor modifications made by us to allow the use of audio files not supported by the original repositories.

The B-RNN training lasted a total of 410 Epochs, ending due to early stopping. We trained the model only once, with no additional retraining. Due to the nature of the recursive networks, this duration for training was expected.

3.4. Estimating tempo

As stated in Section 3.1, the whole musical piece spectrogram is cropped before being fed to the neural network. Thus, the network will estimate the local tempo of each excerpt picking the class with the greatest activation.

To estimate the tempo for the whole track (global), multiple output activations are calculated using a sliding window with half-overlap (hop-size of 128 frames, approximately 5.96 s). The activations are averaged class-wise and the tempo class with the greatest activation is picked as well. This methodology is replicated from [14].

4. Evaluation and results

Following the training stage criteria, the selected datasets for the neural network evaluation step were ACM, GiantSteps Tempo, GTzan and part of Groove (10%), totaling 3,335 different pieces.

To evaluate the performance, we first need to define what is accuracy in tempo estimation. The most objective concept is to estimate exactly the same value (an integer number such as 80 bpm). However, the difference between songs with close tempo values (e.g. 83 bpm and 85 bpm) can be practically imperceptible to human ears, so a small margin of error must also be taken into account, since it does not compromise the understanding of the music rhythm. As such, more than one level of accuracy for the measurement (secondary accuracies) are usually introduced in tempo estimation studies [30].

It is also common for machines to end up interpreting the value of tempo as 2-3 times greater (or smaller) than the time perceived by humans, a phenomenon known as tempo octave error [31], which is also a research subject in MIR. In fact, human beings themselves often disagree as to the value of tempo perceived by each one [30]. One person might, for example, interpret the tempo of a song as 60 bpm, while another one interprets it as 120 bpm. Therefore, even though the best performance would be as close as possible to the true tempo value, it is also of interest to simulate human hearing and its inaccuracies, without discarding multiples of the original tempo.

We defined some support variables to estimate the accuracy of tempo detection (in bpm) by the neural network. 3 types of accuracy were chosen: Accuracy0, Accuracy1 and Accuracy2. It can be said that Accuracy0 and Accuracy1 are the ones that represent the best performance as they are closer to the original absolute value; Accuracy2 represents a performance that can be

considered good enough (and could be refined in future work, not in the scope of this paper). These accuracies refer to the tempi values estimated by the neural networks for one or more musical pieces.

Accuracy0 is the accuracy rate for tempo values (rounded to the nearest whole number) identical to the ground truth label; Accuracy1 accepts estimations with a deviation of $\pm 4\%$ from the ground truth label; and Accuracy2 accepts values two or three times greater (or smaller) than the ground truth label, plus a $\pm 4\%$ margin of error.

Dataset	B-RNN	Sch1 [14]	Sch2
ACM [17]	33.0	40.6	39.3
Groove	58.1	37.2	60.6
GiantSteps Tempo [19]	15.7	27.6	27.7
GTZan[20]	25.2	36.9	30.5

Table 2: Accuracy0 results for the compared models.

Dataset	B-RNN	Sch1 [14]	Sch2
ACM [17]	72.1	79.5	73.8
Groove	76.7	62.8	72.1
GiantSteps Tempo [19]	69.3	64.6	83.0
GTZan[20]	62.0	69.4	64.6

Table 3: Accuracy1 results for the compared models.

Dataset	B-RNN	Sch1 [14]	Sch2
ACM [17]	93.1	97.4	96.5
Groove	95.4	86.0	93.0
GiantSteps Tempo [19]	86.3	83.1	92.5
GTZan[20]	85.2	92.6	91.0

Table 4: Accuracy2 results for the compared models.

These variables were inspired by the earlier study [29], as well as [30] and [14], from which the margin of error percentage ($\pm 4\%$) was replicated. The criteria for each accuracy were chosen not only for they are suitable for an accuracy analysis, but also to facilitate a comparison of the results with a state-of-the-art model. Although we are performing evaluation objectively/quantitatively here, due to the nature of tempo perception, evaluation for tempo estimation can be a bit tricky, and further insights on this matter can be found in [32].

The results for the B-RNN are compared to those presented by Schreiber and Müller's [14] CNN (convolutional neural network), which we will refer to as

Sch1. Since Sch1 presented state-of-the-art results and it was the first not to recur to external modules in tempo estimation, we will be including their results in the analysis; for a comparison of neural networks-only vs. neural networks with external modules, it is recommended to read their original paper [14].

For comparison purposes, the CNN model from [24] was reproduced and trained from scratch with the same datasets and hyperparameters used for the B-RNN of this paper. We will refer to this retrained model as Sch2. This retraining was done as an effort to avoid a possibility of bias in estimating percussion tracks (since Sch1 was not trained with this particular type of dataset).

Tables 2, 3 and 4 each show the percentage of correct estimations for Accuracy0, Accuracy1 and Accuracy2, respectively. Best results for each dataset are in bold.

Before discussing the results, it is worth noting that the CNN has way less parameters than the B-RNN. The CNN is a "simpler" yet effective way of analyzing visual information [22], while B-RNNs present quite more complex layers. Reducing the number of parameters of the B-RNN proposed here could be a good step to test for further improvement in the performance evaluation.

The B-RNN presented better results than Sch1 [14] in all accuracies for the Groove dataset, and better results in Accuracy1 and Accuracy2 for the GiantSteps Tempo dataset. As previously stated, Sch1 was not trained with the Groove dataset.

Nevertheless, B-RNN's performance on the Groove dataset (which contains only drum-line audios) was superior to that of CNNs: of the three types of accuracy established, B-RNN had the best result in two. This can suggest that B-RNNs might detect temporal dependencies better than CNNs on percussion-only tracks. Still, more research should be done on this matter for stronger evidence. Furthermore, Groove was the dataset that received the best accuracies for the models in this experiment.

The Groove dataset had relatively high Accuracies 0, 1 and 2 values for the B-RNN and Sch2 models, which were trained with percussion pieces. In fact, it was the dataset that presented the highest Accuracy0 values, and the only dataset in which B-RNN surpassed Sch1 and even Sch2 (which was also trained with Groove) in accuracy performance, as mentioned already in this section. As the only percussion dataset, containing 443 tracks — that is, only 3.53% of the total pieces —, we had expected the networks would not perform very well compared to the multi-instrumented datasets (which are 12,107 tracks, 96.47% of the total). Obviously, the possibility of bias should be considered since part of the dataset was used for training and the other part for the assessment itself.

Some of the B-RNN's inaccuracies on Groove were a 145-bpm estimation provided for a 290 bpm song. 290 bpm is a very unusual tempo, not only for the training dataset distribution but for music as a whole, and that might have contributed to the choice of 145 instead of 290. Another estimation error was the attribution of 84 bpm to a

track that has 60 bpm, but which has an unusual metric of 6/8, which may have contributed to the error; yet, for another song in 6/8 metric the detection was accurate enough (a 70-bpm track was rated 140 bpm, double the real value).

The highest Accuracy0 value, after Groove, is for the ACM dataset. It makes sense since most of its content is pop and pop-rock songs, which are usually well-defined pieces with few tempo changes. Accuracy2 had the highest values for all Accuracies, which is expected, since it is the least “rigorous” secondary accuracy. Still, since it exceeded 85% accuracy for all datasets (except for the SMC, which was designed to be challenging), it is possible to say that the B-RNN model presents satisfactory results, consistent to human hearing.

It is noticeable that, despite the two CNNs (Sch1 and Sch2) having exactly the same architecture, their performances presented different accuracies for the same dataset. This suggests that the choice of datasets can be decisive for the performance of neural networks — something that is already well recognized in the deep learning field — and that the lack of greater amounts of music available for training can lead to a surprising difference in performance. Therefore, it could prove useful to experiment with different datasets and hyperparameters in future works.

5. Conclusion

In this paper we performed the modeling, training and evaluation of a neural network model (B-RNN) which is capable of estimating the tempo (in bpm) of a musical piece. Providing a piece’s Mel spectrogram as input, the output is a class as the estimation of tempo in bpm (beats per minute). We compared the results of the B-RNN with a CNN model considered state of the art [14], and with a replica of [14] which we retrained with the same datasets and parameters as the B-RNN for the purpose of evaluating the results (aiming to reduce bias). We proposed an analysis on the use of percussion-only dataset for training and the possibility of its use improving accuracy on tempo estimation.

Based on the results, it was noticed that the tempo estimation, overall, was more accurate for the percussion-only dataset. This suggests that tempo estimation can be more accurate over percussion-only tracks, but it is necessary to consider that only one dataset of this type was found available for the experiment, which was also used for both training and evaluation steps. Therefore, it is likely that the results could be biased in this direction. Further experiments should be made with more percussion datasets in order to gather stronger evidence. Still, the amount and variety of tracks in the Groove dataset (the only percussion dataset used) was considered good enough to be the first step on this kind of study. Better results for percussion pieces does not imply that the networks cannot estimate tempo accurately for pieces without such instrument, as long as they have a well-defined global temporal structure.

The B-RNN model also presented very satisfactory results for the Accuracy2, which exceeded 85% of correct answers for almost all datasets. The values of Accuracy0 and Accuracy1, in general, for both B-RNN and CNNs, do not exceed 80%. Since these Accuracies are relevant to the understanding of music and its rhythm, it would be interesting to search for ways to refine this precision, so as to identify the correct multiple value of tempo. The B-RNN also had a better overall performance on the percussion-only dataset, which could suggest that recurrent networks identify temporal dependencies better than CNNs on percussion-only tracks. But further experiments should be done with more datasets for stronger evidence.

None of the models presented superior performance for absolutely every dataset, which suggests that the performance of a neural network in the estimation of tempo still depends a lot on the datasets that are selected for training, as well as the number of pieces used for training. Also, both original CNN model from [14] and its retrained replica presented quite different accuracies for the same dataset, which also enforces such suggestion, also possibly indicating that efforts in constructing more public dataset for research could be fruitful. Still, further evidences [25] [15] have also been suggesting that ANNs might deliver better results in tempo estimation than traditional algorithms, which could (and should) stimulate more research in this field.

For future work, it would be interesting to experiment other hyperparameters for the network trainings; as well as to refine the B-RNN model, creating new layers for better signal processing, or even using B-LSTMs layers. It would also be interesting to create more datasets containing percussion lines-only, in order to reduce the possibility of bias in the assessment. Further experiments could be done to assess whether B-RNNs could be a better choice to identify temporal dependencies for percussion-only tracks.

References

- [1] SCHEDL, M. *Automatically extracting, analyzing, and visualizing information on music artists from the World Wide Web*. PhD Thesis. Johannes Kepler University, Linz, 2008.
- [2] GÓMEZ, E. et al. Music Information Retrieval: Overview, Recent Developments and Future Challenges. In: *17th International Society for Music Information Retrieval (ISMIR) Conference*, 2016, New York.
- [3] ALONSO, M.; DAVID., B.; RICHARD, G. Tempo and beat estimation of musical signals. In: *5th International Society for Music Information Retrieval (ISMIR) Conference*, 2004, Barcelona.
- [4] HÖRSCHLÄGER, F. et al. *Addressing Tempo Estimation Octave Errors in Electronic Music by Incorporating Style Information Extracted from Wikipedia*. Available on: <https://www.ifs.tuwien.ac.at/~knees/publications/hoerschlaeger_etal_smc_2015.pdf>. Access in: 15 ago. 2020.

- [5] HAINSWORTH, S.; MACLEOD, M.; Particle filtering applied to musical tempo tracking. *EURASIP J. on Applied Signal Processing*, v. 15, p. 2385-2395, 2004.
- [6] BÖCK, S. *Onset, Beat, and Tempo Detection with Artificial Neural Networks*. ca. 2010. Available on: <<http://mir.minimoog.org/sb-diploma-thesis>>. Access in 23 ago. 2019.
- [7] BERRY, W. *Structural Functions in Music*. New Jersey: Prentice Hall, 1976. 447p.
- [8] QUINN, S.; WATT, R. The Perception of Tempo in Music. *Perception*, v. 35, p. 267-80, 2006.
- [9] SCHREIRER, E.D. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, n. 103, v. 1, p. 588-601, 1998.
- [10] BÖCK, S.; KREBS, F.; WIDMER, G. Accurate Tempo Estimation based on Recurrent Neural Networks and Resonating Comb Filters. In: *16th International Society for Music Information Retrieval Conference*, Madrid, 2015.
- [11] GKIOKAS, A.; KATSOUROS, V.; CARAYANNIS, G. Reducing tempo octave errors by periodicity vector coding and SVM learning. In: *13th International Society for Music Information Retrieval Conference*, 2012, Porto.
- [12] HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, v. 9, n. 8, p. 1735-1780, 1997.
- [13] BÖCK, S.; SCHEDL, M. Enhanced beat tracking with context-aware neural networks. *Proceedings of the 14th International Conference on Digital Audio Effects*, p. 135-139, set. 2011.
- [14] SCHREIBER, H.; MÜLLER, M. A Single-Step Approach to Musical Tempo Estimation Using a Convolutional Neural Network. In: *19th International Society for Music Information Retrieval Conference (ISMIR)*, 2018, Paris.
- [15] BÖCK, S.; DAVIES, M.E.P. Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation. *Proceedings of the 21st ISMIR Conference (International Society for Music Information Retrieval)*. Montreal, Canada, p. 574-582, 2020.
- [16] BÖCK, S.; DAVIES, M.E.P.; KNEES, P.; Multi-task learning of tempo and beat: learning one to improve the other. *Proceedings of the 20th ISMIR Conference*, 2019.
- [17] PEETERS, G.; FLOCON-CHLOET, J. Perceptual tempo estimation using GMMregression. *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies (MIRUM)*, p. 45-50, 2012.
- [18] MARCHAND, U.; PEETERS, G. Scale and shift invariant time/frequency representation using auditory statistics: application to rhythm description. In: *IEEE International Workshop on Machine Learning for Signal Processing*, Salerno, set. 2016.
- [19] KNEES, P. et al. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, pp. 364-470, 2015.
- [20] TZANETAKIS, G.; COOK, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, v. 10, n. 5, p. 293-302, 2002.
- [21] RAFFEL, C. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. PhD Thesis, Columbia University, New York, 2016.
- [22] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep learning*. Cambridge: MIT Press, 2016.
- [23] MÜLLER, M. Fourier Analysis of Signals. In: MÜLLER, M. *Fundamentals of Music Processing*. Switzerland: Springer International Publishing, 2015. p. 39-57.
- [24] MONTECCHIO, N.; ROY, P.; PACHET, F. The Skipping Behavior of Users of Music Streaming Services and its Relation to Musical Structure. *arXiv:1903.06008*, 2019.
- [25] SCHREIBER, H.; MÜLLER, M. Musical Tempo and Key Estimation using Convolutional Neural Networks with Directional Filters. *Proceedings of the 16th Sound and Music Computing Conference*, 2019.
- [26] SCHUSTER, M.; PALIWAL K.K. Bidirectional Recurrent Neural Networks. *IEEE Transactions on Signal Processing*, v. 45, p. 2673-2681, 1997.
- [27] CLEVERT, D.A.; UNTERTHINER, T.; HOCHREITER, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *arXiv:1511.07289v5*. 2015.
- [28] GRAVES, A.; MOHAMED, A.R.; HINTON, G. Speech recognition with deep recurrent neural networks. In: *Acoustics, speech and signal processing (icassp)*, 2013.
- [29] GOUYON, F. et al. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, v. 14, p. 1832 – 1844, 2005.
- [30] HÖRSCHLÄGER, F. et al. *Addressing Tempo Estimation Octave Errors in Electronic Music by Incorporating Style Information Extracted from Wikipedia*. Available on: <https://www.ifs.tuwien.ac.at/~knees/publications/hoerschlaeger_et_al_smc_2015.pdf>. Access in: 15 ago. 2020.
- [31] WU, F.H.F. (2015). Musical tempo octave error reducing based on the statistics of tempogram. In: *23rd Mediterranean Conference on Control and Automation (MED)*. 2015.
- [32] SCHREIBER, H.; URBANO, J.; MÜLLER, M. Music Tempo Estimation: Are We Done Yet?. *Transactions of the International Society for Music Information Retrieval*, p. 111–125, 2020.

An interplay between genre and emotion prediction in music: a study in the Emotify dataset

Leonardo Vilela de Abreu Silva Pereira^{1*}, Tiago Fernandes Tavares^{1,2}

¹School of Electrical and Computer Engineering – University of Campinas
Av. Albert Einstein, 400, Campinas, SP

²Interdisciplinary Nucleus for Sound Studies
Rua da Reitoria, 82, Campinas, SP

l220198@dac.unicamp.br

Abstract. Automatic classification problems are common in the music information retrieval domain. Among those we can find the automatic identification of music genre and music mood as frequently approached problems. The labels related to genre and mood are both generated by humans, according to subjective experiences related to each individual's growth and development, that is, each person attributes different meanings to genre and mood labels. However, because both genre and mood arise from a similar process related to the social surroundings of an individual, we hypothesize that they are somehow related. In this study, we present experiments performed in the Emotify dataset, which comprises audio data and genre and mood-related tags for several pieces. We show that we can predict genre from audio data with a high accuracy; however, we consistently obtained low accuracy to predict mood tags. Additionally, we tried to use mood tags to predict genre, and also obtained a low accuracy. An analysis of the feature space reveals that our features are more related to genre than to mood, which explains the results from a linear algebra viewpoint. However, we still cannot find a music-related explanation to this difference.

1. Introduction

Each person relates differently to specific music pieces, that is, the same piece can trigger cathartic euphoria for some, while provoking overwhelming negative sensations for others. Sensations related to musical experiences are often regarded to as emotions and feelings, but music is also related to a sense of belonging to particular social groups, whose identity merges to that of its musical aesthetics. This dialogue fosters a process in which musical choices in instrumentation, rhythms, and timbres often give rise to music genres [1].

The idea that some emotions are inherent to music were common in Europe around the XVII Century, and gave origin to the Doctrine of Affects, which directly relates particular musical structures to corresponding affects. This idea was later contested by Hanslick [2] in the XIX Century, who proposed decoupling the ideas of beautiful and the feelings related to music. Hanslick's [2] argument is that music can trigger emotions in individuals, but each one's experience is subjective and particular.

This means that music does not carry feelings by

itself, the same way that spoken words do not carry feelings by themselves. Rather, feelings emerge within the individual's brains, and they are highly dependent on each one's growth and life story.

Nevertheless, humans learn adequate forms for expressing their emotions during their childhood [3]. This process is intertwined with the development of each person's communication skills, which allows expressing emotions (inner states) as feelings (externalized states). Thus, feelings are constructions that vary for each individual, but also tend to be somewhat uniform within a specific culture or ethnicity.

Like feelings and their expression, music and the process of making and listening to music are also social constructs. The responses, perceptions and uses of music are learned by individuals based on the social group they belong to [1]. With time, these identities can spread beyond small communities and become well-known, commercial music genres [4].

Feelings and music genres come from similar processes. They are learned by individuals in a dialogue with their communities, and some of their aspects can become more common, possibly being considered a part of the communities themselves.

Both feelings and musical genres can be important pieces of information about music, allowing cataloguing, analyzing, and searching within databases. For this reason, the automatic identification of feelings and of musical genres have both been broadly studied in Music Information Retrieval (MIR).

Automatic music genre identification is an MIR task that has been studied for decades now [5, 6]. It works by calculating features from audio signals, thus mapping audio files into a $\mathbb{R}^N$ vector space. Then, classifiers can use labeled data to find an optimal class-related partition within this vector space.

Music-related emotions can be predicted using a similar process, in which audio is mapped into a $\mathbb{R}^N$ vector space [7]. Interestingly, this mapping can be performed using the same features used for music genre classification [5], and classifiers use emotion-related labels to partition the vector space. This further indicates a similarity between genre- and affective-related labels in music.

*Supported by CNPq.

The correlation between genre and affective labels has been observed in work by Aljanaki et al. [8]. They observed that some music genres are more often related to specific affective labels. However, this does not happen for all affective labels.

In this study, we investigate this correlation from a machine learning point of view. More specifically, we investigate whether affect-related predictions can be useful to predict genre, and vice-versa. The goal of this study is to understand whether, and to what extent, music genres and music affects are simply different perspectives of the same phenomenon.

An important aspect of music labels is that humans often disagree among themselves regarding the genre [5], emotion [8], and even the subject [9] of songs. For this reason, we do not aim at achieving high prediction accuracies. Instead, we focus on finding explanations about the behavior of classifiers and feature spaces.

In the next section, we show the data and methods used in this investigation.

2. Methods

Currently, there are many readily-available datasets for music genre identification and music emotion recognition, as well as a myriad of solutions for these tasks. In this work, we did not focus on developing a new dataset or a new method; rather, we aimed at gathering evidences on the interplay between the genre- and affect-related labels from a machine learning perspective. We discuss the dataset in Section 2.1 and the experiments in Section 2.2.

2.1. Dataset

In this work, we used the Emotify dataset [8]. It contains 400 pieces gathered from the Magnatune dataset. Music genre labels were assigned by the recording companies, generating a balanced dataset with Rock, Pop, Classical, and Electronic genres.

Each piece was rated in a user study in which users assigned affective labels to songs. These labels came from the Geneva Emotional Music Scale (GEMS) model [10], which consists of assigning a binary value to 9 categories: Amazement, Solemnity, Tenderness, Nostalgia, Calmness, Power, Joyful activation, Tension and Sadness.

We further processed the dataset in order to find single tags for each song, as follows. First, we calculated $\mu_{s,k}$, which is the number of users that rated song s as positive for emotion k divided by the total number of ratings for song s . Then, we calculated the median m_k of $\mu_{s,k}$ along all songs for emotion k . Finally, we assigned the value 1 for emotion k in song s if $\mu_{s,k} > m_k$, and 0 otherwise.

2.2. Experiments

We performed a series of experiments using the Emotify dataset. First, we used a classification process to predict genre and affective labels directly from audio signals, as shown in Section 2.3. Then, we experimented using the

results of the affective label predictions as feature vectors in the problem of identifying music genre, as discussed in Section 2.5.

2.3. Genre and emotion prediction from audio

The audio-based classification used a pre-trained VGG19 neural network [11]. The features generated by the VGG19 were yielded to a simple pipeline comprised of a scaler and a K-Nearest-Neighbors classifier. In our experiments, we used $k = 5$ neighbors. We randomly used 70% of our data for model fitting and 30% for testing.

The PCA projection of our testing data, shown in Figure 1, highlights that the VGG19 was able to find features that strongly separate music genre within this dataset. Unsurprisingly, the prediction accuracy for genre in the testing set was 100%, as displayed in the confusion matrix shown in Figure 2.

2.4. Experiment I - Audio input predicts genre

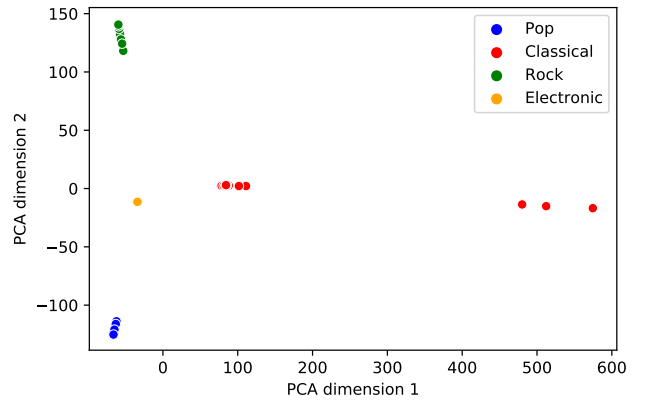


Figure 1: Audio feature space PCA projection colored by ground-truth genre labels.

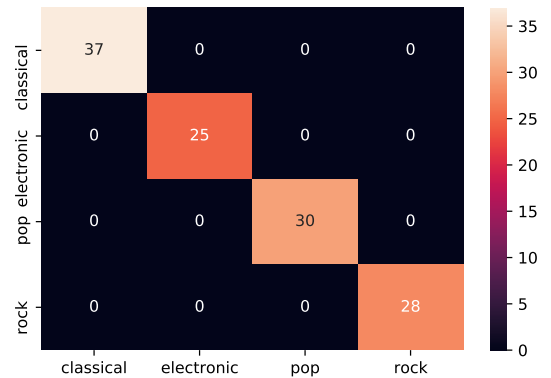


Figure 2: Confusion matrix for audio input, genre output.

We highlight that this does not mean that VGG19 is an ideal audio feature extractor. Rather, these results can be a reflection of characteristics that are specific of the Emotify dataset.

Using the same experimental setup, we conducted experiments for each one of the affective labels. The accuracy for each label is shown in Table 1. As we can see, the prediction accuracy is significantly lower than in the case of genre identification.

Table 1: Accuracy for emotion prediction using audio input.

Emotion	Accuracy
Amazement	0.52
Solennity	0.67
Tenderness	0.67
Nostalgia	0.63
Calmness	0.57
Power	0.62
Joyful Activation	0.52
Tension	0.63
Sadness	0.43

Figure 3 shows the confusion matrices related to each one of these emotions. They further confirm that predicting affective labels in this dataset is a harder task for machine learning, when compared to genre prediction.

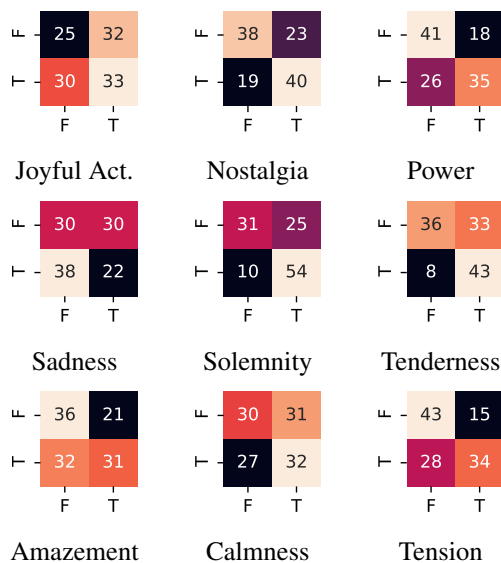


Figure 3: Confusion matrices for each emotion in the emotion prediction from audio experiment.

Although genres and affective labels come from a dialogue between individuals and their societal surroundings [3, 1], and that similar feature sets have been used to predict both genre and affective labels in the past [5, 7], we observed that the VGG19 features are much more effective to predict genre than to predict emotion. We add that previous analysis in the Emotify dataset [8] have encountered some correlations between the most frequently chosen emotion tags and music genres. However, our system had an obviously different performance when dealing with genre and when dealing with affective labels.

In the next experiments, we use GEMS ratings as

feature vectors to predict genre.

2.5. Genre prediction from emotion space

The experiments shown in this section use GEMS ratings to predict music genre. We performed two of such experiments: the first used the ratings estimated in the data pre-processing stage discussed in Section 2.1, and the second used the estimated probabilities yielded by the KNN classifier discussed in Section 2.3. These values were yielded to a further KNN classifier that predicts genre-related labels, and, as with the previous experiment, we used $k = 5$ neighbors and a 70%/30% train/test data split.

The dataset GEMS ratings spans an interesting vector space, whose PCA projection is shown in Figure 4. As we can see, although it seems that each genre has a greater tendency of occupying a different region in the vector space, there is a great overlap between the regions. It is hard to determine if this is a characteristic of music genres or if it is a specific bias of this dataset.

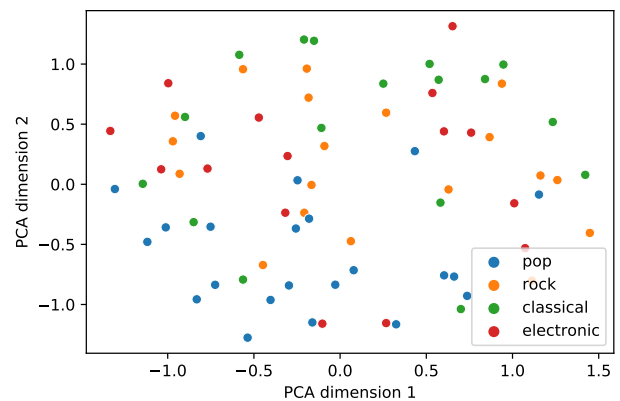


Figure 4: PCA projection of vector space spanned by GEMS ratings in the dataset.

The prediction experiment results, shown in Figure 5, indicate a greater accuracy for the Pop genre. This can be linked to the predominance of Pop songs in the lower left of the PCA projection. Likewise, this can be either a characteristic of music genres or of this specific dataset.

When we span the emotion vector space using prediction probabilities from the audio-based classifier, we have a somewhat different result. The space spanned by these predictions, shown in Figure 6, indicates an even higher overlap between the regions occupied by each genre when compared to the previous result. The confusion matrix for this experiment, shown in Figure 7, indicates a high tendency of polarization towards the Classical genre.

3. Discussion

In this work, we experiment on the prediction of genre and emotion labels from audio. After that, we try to use emotion-related labels to span a feature space that allows to predict genre. We perform experiments in the Emotify

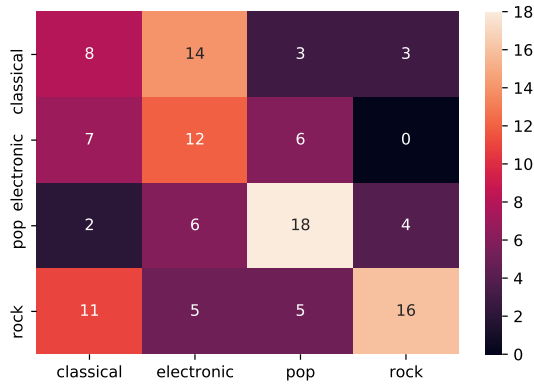


Figure 5: Confusion matrix for genre prediction using GEMS ratings.

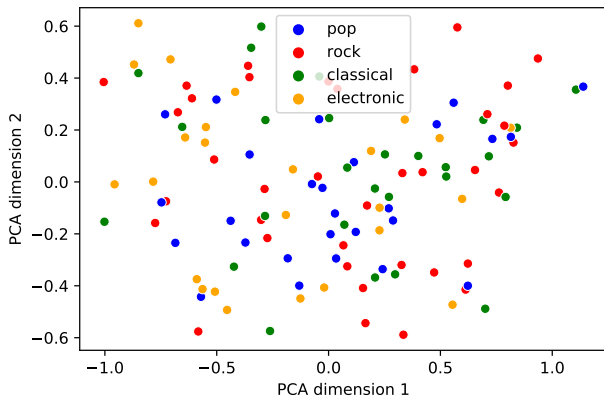


Figure 6: PCA projection of vector space spanned by GEMS ratings predicted from audio.

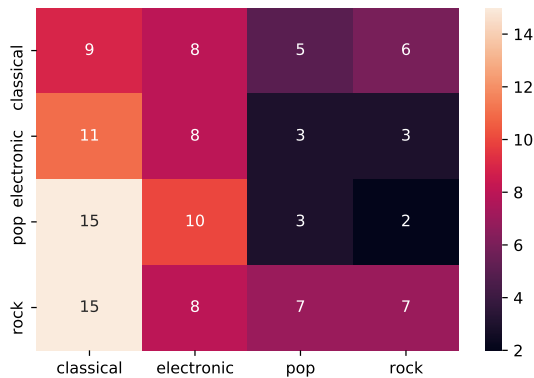


Figure 7: Confusion matrix for genre prediction using GEMS ratings predicted from audio.

dataset, which contains both genre and emotion-related labels.

We used the idea that both music genres and music-related emotions come from a dialogue between the individuals and their surrounding society. Music genres are

related to the inter-social affective relationships that arise among individuals, and emotions are, as well, related to this shared individual and social identity. This idea is corroborated by studies in psychology [1] and neuroscience [3].

However, we observe that the prediction accuracy for music genre is evidently higher than that for emotion prediction. This means that, at least for our dataset, genre and emotion labels are not different viewpoints of the same phenomenon. Rather, they seem to derive from different underlying processes.

One important characteristic of the dataset is that genre labels were provided by recording companies, that is, they have a meaning related to their commercialization. On the other hand, the emotion-related tags were provided by humans, thus are related to subjective experiences [8]. Also, they use a specific model – the GEMS scale – which can also cause distortions that impact the prediction accuracy.

This accuracy, albeit relatively low when compared to genre prediction, do not mean that the experiment failed. On the contrary, they can be a result of the subjectivity of the emotional experience when listening to music. Such subjective causes humans to disagree among themselves on the GEMS ratings [8], henceforth machine learning algorithms cannot possibly achieve a high accuracy, as this would mean a super-human capacity of predicting emotions, that is, an obvious algorithmic bias.

Next, we used emotion ratings to predict music genre. This experiment had a visibly lower accuracy when compared to genre prediction from audio features. This can be due to a series of elements.

One possibility is that genre and emotion are inherently uncorrelated. Although this was observed in our dataset to some extent, it seems clear that some music genres tend to have associated affective meanings – for example, Punk music is commonly associated with teenage anger, whereas Reggae brings a much calmer mood. However, it can be the case that the procedure to produce the genre and emotion labels are uncorrelated to the social and cultural meanings often attributed to them.

Another important aspect is that the audio feature space was spanned using a VGG19, which produces vectors with around 10^7 features, while the emotion space uses 9 dimensions. Hence, the VGG19 is projecting audio excerpts into a very high-dimensional vector space, in which we observed a high separability for musical genres. However, the VGG19 was trained to perform image classifications, that is, we are using a transfer-learning technique.

This could indicate that the VGG19 features are unrelated to genre-specific musical style, and are somehow exploiting audio characteristics that derive from typical mixing and mastering processes related to each genre. This means that the high accuracy achieved in the genre classification problem could be due to properties of high-dimensional spaces and some coincidences within the

dataset. These conditions disappear when we use a low (9) dimensional space to represent tracks, hence leading to a lower accuracy.

Such reasoning brings forward the importance of the PCA projections. In Figure 4, for example, we can see that music genres overlap in the vector space spanned by GEMS ratings. This projection indicates that the prediction accuracy was not harmed necessarily by the low amount of data; rather, the genre sets are inherently correlated and it can be hard to separate between them.

Nevertheless, we observe in Figure 4 that the overlap between genres is not along the whole emotional space. Also, each genre seems to have a slightly different median in the emotional space. Consequently, we speculate that although music genres are somehow related to affective values, there is enough space within each music genre to compose music with significantly different affective value.

Next, we present concluding remarks.

4. Conclusions

In this article, we compared the prediction of genre and of emotions in music from audio using a transfer-learning technique. We also investigate the prediction of genre from a feature space spanned by emotion-related ratings.

We obtained a high accuracy when predicting music genre from audio features. We believe that this result is due to properties of high dimensional spaces and production biases (mixing and mastering techniques) in the dataset.

Also, we identified that musical genres are located in overlapping regions within the emotional-spanned feature space. This means that, regardless of the dataset size, there can be music from different genres that are related to the same emotion.

Lastly, we highlight that this work does not aim to achieve a higher prediction accuracy. Rather, we propose an analysis aimed at explaining why the prediction methods perform the way they do. We provide this explanation under the light that both genre and emotion labels are subjective.

There are many possible steps to improve our ability to explain the behavior of classifiers in the audio domain. We anticipate that using more concrete concepts, such as key or tempo, as support could lead to results that correlate more to musicological concepts. Such idea poses a vast field for future work.

References

- [1] B. Ilari, “Música, comportamento social e relações interpessoais,” *Psicologia em Estudo*, vol. 11, no. 1, pp. 191–198, Apr. 2006.
- [2] E. Hanslick and G. Payzant, *On the musically beautiful : a contribution towards the revision of the aesthetics of music / Eduard Hanslick ; translated and edited by Geoffrey Payzant*. Hackett Pub. Co Indianapolis, Ind, 1986.
- [3] A. Damasio and G. B. Carvalho, “The nature of feelings: evolutionary and neurobiological origins,” *Nature Reviews Neuroscience*, vol. 14, no. 2, pp. 143–152, Feb. 2013. [Online]. Available: <http://www.nature.com/articles/nrn3403>
- [4] J. C. Lena, *Banding Together: How Communities Create Genres in Popular Music*. Princeton University Press, 2012.
- [5] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.
- [6] J. H. Foleiss and T. F. Tavares, “Texture selection for automatic music genre classification,” *Applied Soft Computing*, p. 106127, feb 2020.
- [7] T. Li and M. Ogihara, “Content-based music similarity search and emotion detection,” in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, 2004, pp. V–705.
- [8] A. Aljanaki, F. Wiering, and R. Velkamp, “Collecting annotations for induced musical emotion via online game with a purpose emotify,” UU BETA ICS Departement Informatica, Tech. Rep., 01 2014.
- [9] A. Dalmora and T. F. Tavares, “Identifying narrative contexts in brazilian popular music lyrics using sparse topic models: A comparison between human-based and machine-based classification,” in *XVII Brazilian Symposium on Computer Music*, São João del Rei, MG, Brazil, sep 2019.
- [10] M. Zentner, D. Grandjean, and K. R. Scherer, “Emotions evoked by the sound of music: Characterization, classification, and measurement,” *Emotion*, vol. 8, no. 4, pp. 494–521, 2008. [Online]. Available: <http://doi.apa.org/getdoi.cfm?doi=10.1037/1528-3542.8.4.494>
- [11] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.

Mandrit: a system for automatic generation of musical visualizations for rhythm analysis

Rute Moura^{1*}, Giordano Cabral¹, Horhanna Almeida¹, Ricardo Scholz¹

¹Music- Centro de Informática/Universidade Federal de Pernambuco
Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária(Campus Recife) – 50.740-560, Recife, PE

rutymaxsually@gmail.com, grec@cin.ufpe.br, hao@cin.ufpe.br, ricardo@marineframework.org

Abstract. Faced with the representational challenge of visually communicating musical information, we developed Mandrit: a system for automatic generation of musical visualizations for rhythm analysis. It uses data extracted from MIDI files to plot graphs for different songs, representing their rhythmic signatures. The system is customizable and presents new ways to visualize pace through different metrics. We aim to encourage the discussion in this field and the development of tools to generate musical visualizations, as a means to help musicians and scholars in musical analysis, teaching, composing and complementing musical performances.

1. Introduction

Humans are visual. Thus, it is reasonable to expect that communicating a message is faster through a visual representation. Music has many representational challenges, as it is rich in abstract, complex and subjective information. In this sense, there is a continuous need for analytical studies in Musicology. Therefore, its representational aspects are associated with "an aesthetic dimension of information, memory, cognition, sensitivity, and knowledge of the musical work" [1].

The theoretical basis of this work is in the interdisciplinary area of Computational Music Visualization, also called *MuVis*. It has a growing community, which held the 1st MuVis Symposium in 2018 [2]. For its wide applicability, MuVis brings together studies from various areas, such as Information Visualization, Musicology, MIR (Music Information Retrieval), Education, and Design. It consists of contributions, documentation, and practical studies, aimed at the development of tools, algorithms, techniques, and methods. It is concerned with musical information and the creation of musical visualizations applied to different contexts, such as visual communication of musical concepts [3], assistance in the performance and to teach music [4], musical analysis [5], or even the creation of compositions and artistic performances [6].

In this work, we emphasize the study of possibilities for visual communication of rhythmic elements. Knowing the potential of software resources - for processing and manipulating data from digital files -, we developed the Mandrit system: a system for automatic generation of musical visualizations for rhythm analysis.

*Supported by FACEPE.

2. Generation of Music Visualizations

During the development process of Mandrit, we carried out a case study applying the Design Thinking (DT) approach, formulated and expanded by IDEO [7]. It helped us to explore the generation of personalized views through a structured process in four steps: problematization, ideation, prototyping, and evaluation. In this article we focus on describing the last two steps: prototyping - the generation of visualizations - and evaluation of the system.

We addressed the complex problem of visually communicate musical information through a user-centered qualitative-quantitative research based on feedback. We held individual sessions and focus groups, in which we collected data, further analyzed. Specialists and beginners in Music and Information Visualization participated in the experiments, collaboratively generating paper prototypes, as shown in figure 1. From these sketches, we were able to glimpse the graphic possibilities and challenges to visually represent and communicate musical information.



Figure 1: Paper prototypes generated by participants of individual sessions and focus groups

We performed workshops and tutorials, seeking to explore the process of creating visual representations of music information. During this phase, we noticed a concern of the participants about representing emotional aspects associated with context, designing the instruments played in the song, and trying to communicate their elementary characteristics. They used terms such as tim-

ing, beats, intensity, silences, pauses, tempo, and expressiveness, directly relating them to graphic features, such as shapes, colors, textures, contours, and fills, seeking to represent what they recognized as musical information to visually explain music.

The participants created paper prototypes, each of which being a possibility of visual representation for musical content, from the user's point of view. In this article, we focus on their most recurrent demands and needs: the analysis of rhythmic elements and the generation of graphs that communicate them, as rhythmic signatures.

2.1. Mandrit Computational Prototyping

Computational prototypes allow the use of technologies and software resources to develop visualizations, exploring data from digital music files. In addition to making it possible to design systems based on the prototypes, it also allows to simultaneously perform simulations and validations with users, what can directly contribute to verify the feasibility of the project or the use of a given technology. To develop musical computational visualizations, we defined a structured prototyping process in three steps, as shown in the flowchart of figure 2.

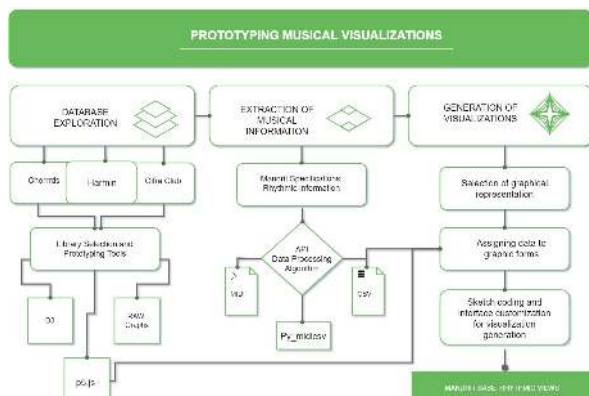


Figure 2: Prototyping process

In the first step, "Database Exploration", we looked for similar existing tools and methods already used to develop musical visualizations. In the second step, "Extraction of Musical Information" (specifically rhythmic information, in Mandrit), we processed the source data to extract meaningful musical information. The last step, "Generation of Visualizations", involved selecting the best graphical representations, with real data, to generate visualizations. Mandrit¹ was developed in the last two steps, and its name references "rhythmic mandalas". Through the attribution of rhythmic information to graphic forms and computational manipulations, it generates three types of graphic models aiming to communicate rhythm.

We carried out tests focused on the automation process and the generation of visualizations. We structured a configurable graphic space to plot the chosen representations according to data extracted from the MIDI files.

¹DOI Mandrit: <https://zenodo.org/record/4118575>

2.1.1. Extracting rhythmic data from MIDI

MIDI files are symbolic musical information. Nevertheless, rhythmic features are not given directly in this format. Thus, we needed to perform a feature extraction step, before generating the graphs. As we sought to develop visual signatures of the rhythm, we initially determined which properties of the rhythm would be assigned to the visualization: the type of instrument performing the song, the data about the bar subdivision, and the number of notes with their respective values. To capture this information, we developed a rhythmic extraction algorithm for MIDI files that transforms, organizes and stores this data in variables in a structured way. Table 1 shows a snippet of data from the song "Bateria04" that we produced to develop the algorithm. Column *X* is the track in which the instrument is played, while column *Y* is the subdivision in which the onsets occurred, in the bar. Finally, column *Total notes* sums up the onsets corresponding to that specific tempo. In other words, in this snippet, we show the sequence of track 2 with its respective values of time and number of notes played.

X	Y	Total_notes
2	0	49
2	1	48
2	2	65
2	3	56
2	4	71
2	5	55
2	6	69
2	7	56
2	8	33

Table 1: Snippet of rhythmic data extracted.

This was used to set up the Mandrit database, which has a collection of more than 60 music files already structured in CSV format and processed according to their rhythmic information. Thus, in order to transpose this information and make it visually communicable with rhythmic signatures, we use the information from each song as input.

Moving on to the visualization generation step, we applied a cyclical validation process and selected some paper prototypes to transform them into graphic models.

2.2. Generating Mandrit Graphics

To generate graphically communicable, automated, generalized visualizations applicable to different songs, we attributed elementary rhythmic information to visual structures and graphic forms.

We carried out practical experiments aided by software resources, i.e., we used computational functions of order, rotation, and translation to arrange the musical parameters in a configurable area with graphical representations - points, lines, circles, polygons, colors, size - and manipulate each visual property in order to associate it to the preprocessed data.

Due to research cutout, the views generated by Mandrit are static and two-dimensional, having the geometric figures and orientations associated with aspects of the metrics of the bars. In the first experiments, shown in figure 3, the results of manipulations and graphical plots allowed us to observe the challenges of visually processing large volumes of data in a temporal sequence. These challenges include graphic pollution, difficulties in following the musical sequence and reading the graph - due to the accumulation of information -, and interactivity limitations to hide or analyze components separately.

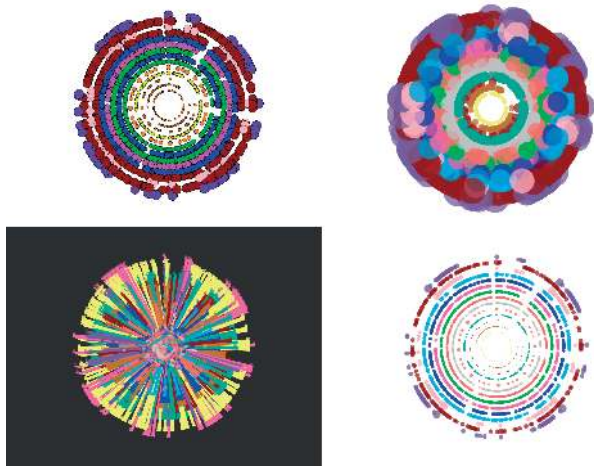


Figure 3: First experiments

Furthermore, we noticed that MIDI files used in these prototypes were not well organized. Their tracks were arranged randomly, without a consistent ordering, neither they presented instrumental descriptions. This turned to be an additional issue when structuring data. In order to quickly overpass this issue, we decided to use MIDI files from Muscore², as it is a platform aimed at professional productions, providing MIDI files structured according to the musical score.

2.2.1. Visualizing the rhythm

Then, we remodeled the visual outputs, seeking to improve visualization and readability, communicate rhythm, avoid graphic pollution, and deliver a more intuitive and understandable graphic result. We selected some default values according to musical parameters and visual structures, as described in table 2.

By associating musical and visual parameters, in addition to valuing the rhythmic signature, we set default values to configure the graphic spaces and present communicable results. These customizable values were based on loyalty tests [8] with user ratings through musical analysis applying an online form to collect qualitative and quantitative data regarding the readability of the views, this step is detailed in section 4. Then, we defined the three types of graphs that would be generated by Mandrit: *Polar Bubble Chart*, *Polar Radar chart* and *Radial Column*.

²Muscore: <https://musescore.com/sheetmusic>

Musical Parameters	Visual Parameters	Generalizations
Amount of notes	Rotation and Translation Geometric shapes: Circles, Polygons and straight segments.	Variation data from matrix of each song
Tracks	Equidistant spacing	Smaller radius = 100; Larger radius = 300;
Temporal cycle and Granularity	Length of circumference	64; or 32; or 8;
Ordering of Notes	Ratio between bubbles; Positions' alignment	Smaller size = 1; Bigger size = 3;
Analysis Window	Reference variation of the measure	2 to 5;

Table 2: Music visualization parameters

The graphs were projected with its elements arranged in a polar area, with angular rotation depending on the subdivisions of the beat of each song. In addition, we applied a color pattern to the tracks, which role is to represent a given quantitative data, transmitting balance, composition, proportion and harmony to the views. The difference between Mandrit's graphics lies in the attribution of geometric shapes, specific to each one of them. The Polar Radar chart generates polygons, while the Radial Column chart draws straight radial lines and the Bubble chart uses circles of different sizes.

With the *Polar Radar Chart*, shown in figure 4, the graphic model is formed by irregular polygons that are traced by varying the amount of notes played at each beat of the musical sequence - and, consequently, according to the stored data of each song -, which are plotted through rotations, vertex connections and considering the time displacement of the measure. This generates the resulting figure, which, depending on the moment in the musical sequence, presents rhythmic accents in the visualization, allowing to easily spot which parts of the song have more onsets, for instance.

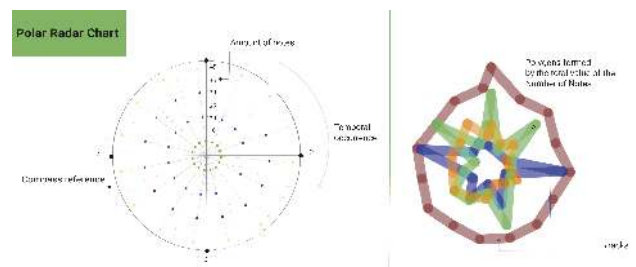


Figure 4: Polar Radar Chart

We emphasize that the polar and the cyclic projections prioritize the temporal aspect of the rhythm and, as a result, they are visualizations of the whole song, with its rhythmic signature. They split instruments hierarchically, based on the amount of onsets, and organize them in the polar orientation, through angular variations, rotating clockwise. According to the subdivision of the measure during the musical sequence, they demarcate the rhythm and refer to a clock, as a metaphor to the temporal aspect, having as reference the representation of rhythm by Toussaint[9].

Seeking to better understand the behavior of temporal data, we generated the *Radial Column* view, shown in figure 5. It allows to separately visualize the temporal sequences of the tracks, i.e., when onsets occur for each track, within the bars. The tempo leaves traces in the form of straight segments rotating from the center, directed like a polar axis, moving in angular rotations exploring the entire area determined by the cyclical visualization.

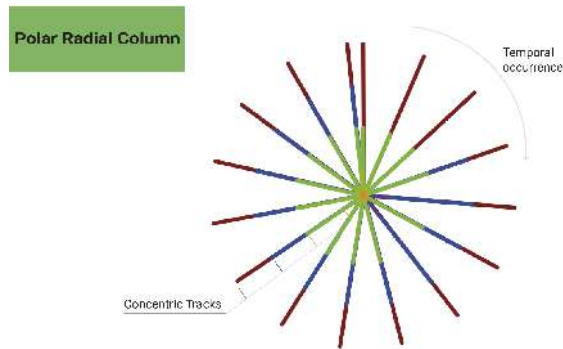


Figure 5: Radial Column Chart

This graph is a minimalist view of temporal data, as it remains grouped, and music intervals and silences become more visible. However, we miss the amount of notes, a crucial information for some analysis.

For this reason, we finally developed the *Polar Bubble Chart*, which groups the amount of onsets and their temporal occurrence in a single geometric shape, a circle. Its area demonstrates the variation in the amount of onsets, and its angular position is determined by the subdivisions of the bar and the length of the temporal sequence.

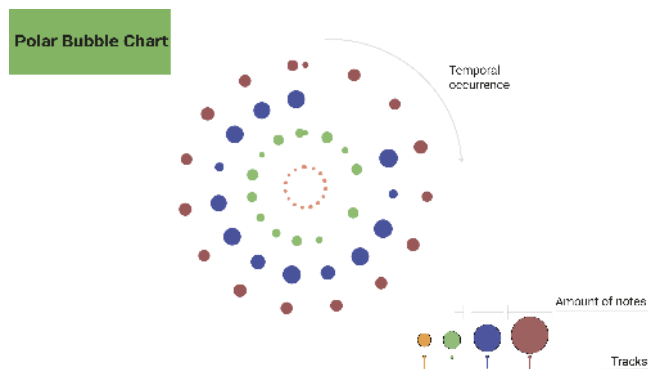


Figure 6: Polar Bubble Chart

The prototyping process led to the graphic models used by Mandrit, which can be customized by manipulating code, if necessary to improve visualization, depending on the volume of data in a given song. Figure 7 shows an example, in which we need to increase the radius of the track view to better occupy the graphic space.

Therefore, visual parameters can be set according to the musical parameters. It is possible to vary the size of the temporal cycle according to the granularity and the analysis window, allowing users to perceive the rhythmic information contained in the visualization, improving

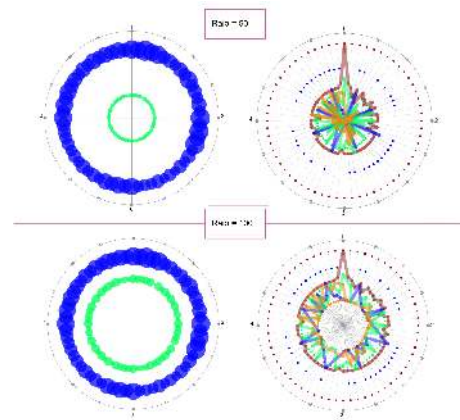


Figure 7: Visual parameters

Prägnanz [10] of the form, and increasing visual organization.

It is important to highlight that during the entire prototyping phase, in addition to the specifications of the graphic projection associated with the musical parameters, we also improved the interface, including subtitles and interactive aspects to enable the selection of the type of visualization, and allowing to hide tracks, in order to analyze them separately. And we have also developed an animated cursor which helps to follow the musical performance, as shown in figure 8.

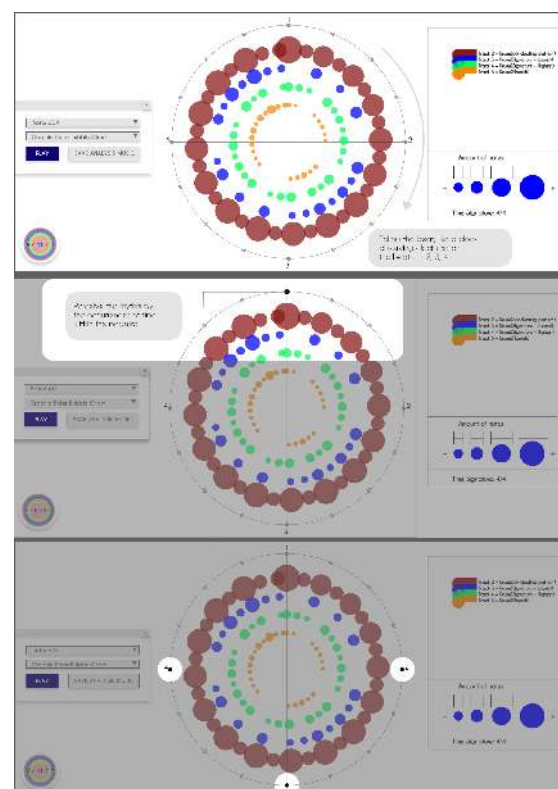


Figure 8: Mandrit tutorial

The cursor follows the rhythm metric according to the music being played and its associated data extracted by Mandrit. With these views, and being able to change

to the three graphic models for the same song, it is possible to identify musical peculiarities in relation to rhythmic aspects as a function of tracks, bar subdivisions and the amounts of onsets. Thus, we carried out comparative musical analysis with rhythm visualizations to assess the system and its communicative usefulness.

3. Musical Analysis

Simultaneously to the prototyping process, we carried out musical analysis, collecting feedback from potential users and Music experts. These analyses helped us to understand how to visually communicate rhythm signatures. When performing several plots to visualize different songs, we used graphic images to make qualitative and quantitative musical analyses regarding visual, musical, and communicative perception of Mandrit's graphics.

The musical parameter of the analysis window is an example of these continuous improvements. It made visual reading and musical accompaniment of the regularity of each bar cycle easier. We simplified and generalized the subdivisions of the beat, with a musical language already established and closer to music professionals, by using timestamps based on the count of rhythm. For example, figure 9 shows songs in different measures - 2/4, 3/4, 4/4, and 5/4 -, with the references of the bar subdivisions plotted in gray lines.

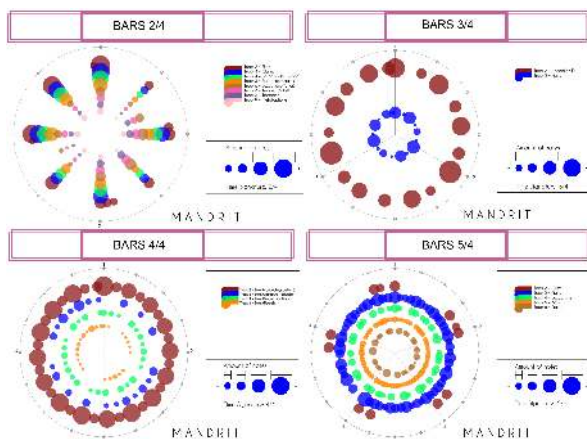


Figure 9: bars

In these plotted songs, we reference their measures by the divisions equivalent to their metrics. For instance, we plotted the song "Asa Branca, by Luiz Gonzaga", in 2/4, by dividing the circumference into 2 beats (180 degrees apart from each other). As well as "João e Maria, by Chico Buarque", in 3/4, and its 3 beats (120 degrees apart) with a production of only two instruments. For a 4/4 example, we plotted a custom file, produced for test purposes, named "Bateria 04" (each beat, 90 degrees apart). And finally a more complex analysis, with the song "Take five, by Dave Brubeck", in 5/4, with 5 beats (72 degrees apart). Thus, the graphic customization varies according to the measure signature of each song loaded.

Mandrit allows performing comparative and discursive analyses over songs with the same time signature,

but having peculiarities, according to their musical genres and visualizations. It also complements the analytical process, aiding it with a visual perspective, and helping to understand rhythmic complexities. For instance, figure 10 shows graphic results of four songs from different musical genres.

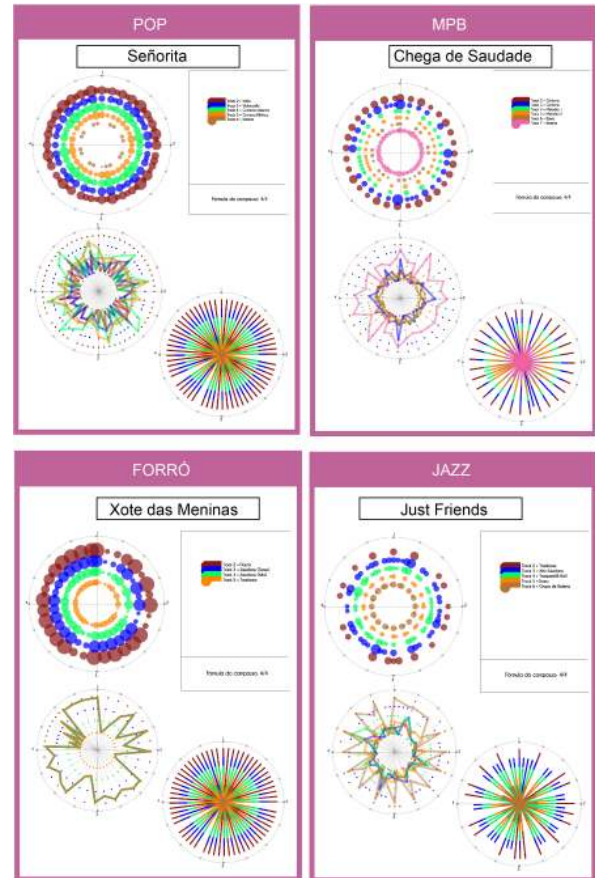


Figure 10: Musical genres

The musical visualizations generated by Mandrit also made it possible to identify macro-structural characteristics of rhythm in Brazilian music, as shown in the graphic results of figure 11. All songs are in 2/4, but their representative signatures can generate different qualitative and interpretive impressions. They show various heterogeneous or homogeneous visual variations, corresponding to their rhythms, instrumental aspects, temporal synchronization, and sequential repetitions performed in the songs, each with its originality.

The song "Aquarelas do Brasil", by Ary Barroso, one of the most recorded songs in Brazil, considered MPB classics, has a cyclic symmetrical signature along the musical performance, meaning that it has a proportion and balance in the variation of the amounts of onsets. This is demarcated in the quadrants of the polar area, enriched by several notes throughout the temporal cycle, and also by the greater presence of values at certain times represented by the circles shown in larger dimensions. Similar to the second song, "Rosa Amarela", by Villas Lobos, both structured around the theme of popular music, they emphasize culture and customs, and regarding visualization, they ex-

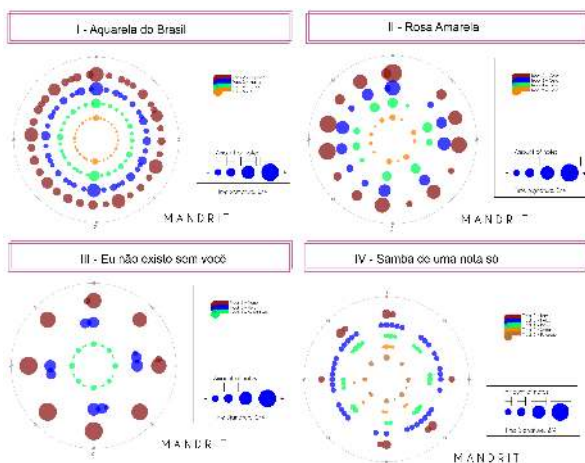


Figure 11: 2/4 Song Analysis

plore the graphic area filled with musical information.

In the case of the second song, we used a file with the complete arrangement, with all the Cello tracks. This raised questions about how to analyze songs more accurately, for instance, in productions with instruments with the same timbre. As the visualization demonstrates, the instruments perform the same rhythm, but micro variations are perceived, considering that the different tracks are equivalent to the expressiveness of each musician. Thus, in this visualization, we notice factors such as synchrony and a remarkable search for the resemblance musical aesthetic, familiar to a symphony or concert, in addition to the fact that the instrumental execution of the cello provides the tempo and enriches the musical dynamic. In the third and fourth songs - "Eu não existo sem você" and "Samba de uma nota só", by Tom Jobim, compositions by the same artist that have a focus on words and melody, both with the same metric, 2/4, we observe a clean visualization in keeping with its simple beat rhythm.

4. Results and Discussion

In this section, we show quantitative data related to the results of musical analysis performed against Mandrit visualizations. We interviewed 17 potential users, who filled an online evaluation form. The form included a textual presentation of Mandrit application, tutorials, videos, and graphic models, as well as visualization of 10 songs: "Baião and Xote das Meninas" (Luiz Gonzaga), "Maracatu Atômico" (CSNZ), "Girl from Ipanema" (Tom Jobim), "Frevo Mulher" (Zé Ramalho), "Take Five" (Dave Brubeck), "Yesterday" (The Beatles), "Machine Gun" (Jimi Hendrix), "Just Friends" (John Klenner), and "Señorita" (Shawn Mendes and Camila Cabello).

We carried out an assessment of Findability [11] of musical information associated with the visual structures of the graphics, questioning the participants' impressions about these aspects and their cognitive perceptions while reading the graphic visualizations. Through this, we gathered quantitative-qualitative data concerning the advantages and disadvantages of visualizations for rhythm

analysis, and suggestions and feedback regarding continued improvements of Mandrit system.

As shown in figure 12, 58.8% of the public (10 participants out of 17) perceive and identified the "occurrences of notes" and "metrics". Only 3 (17.0%) participants recognized the time signature by the views and 4 (23.0%) were able to perceive accents, metrics, duration of silences, and the time signature itself. Two other participants were able to perceive the sequences and the notes played in syncopation and only one of them was able to understand all the musical information in the visualization.

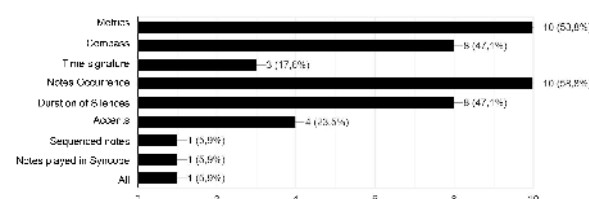


Figure 12: Music Information Findability

There is a notable difficulty for participants to understand musical properties in the first contact with Mandrit visualizations. However, after some time they develop the ability to grasp the visualization and understand its musical aspects, and keep track of rhythm in the musical performances. In general, participants considered a complex exercise to understand the rhythm in static views, due to the amount of information in the graphic area, making it difficult to read and follow the rhythm. Figure 13 shows a comparative evaluation between the three graphs generated by Mandrit: Polar Bubble Chart (Graph A), Polar Radar Chart (B) and Radial Column Chart (C).

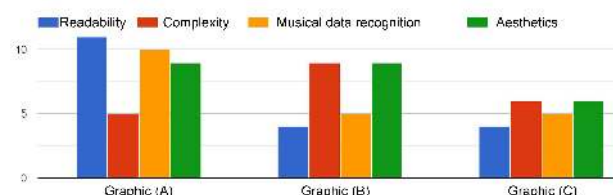


Figure 13: Comparative Evaluation

The Polar Bubble Chart (A) was considered legible, not very complex, having good aesthetics and allowing to recognize musical information. The Polar Radar Chart (B) has lower readability, being more complex to analyze and consequently more difficult to recognize musical information, despite its high aesthetics valuation. Finally, the Radial Column (C) was rated as having lower relevance as a whole. One of the participants reported that its cyclical bar chart becomes very confusing, lowering readability. Despite its aesthetics having been valued by some of the participants for its visual cleanliness, they considered it a little complex and could not recognize musical information well.

When asked which of the three views generated by Mandrit were more communicative/informative, 58.8% of the participants selected the Polar Bubble Chart (A), as

shown in figure 14, arguing that association between the graphic and musical information was better.

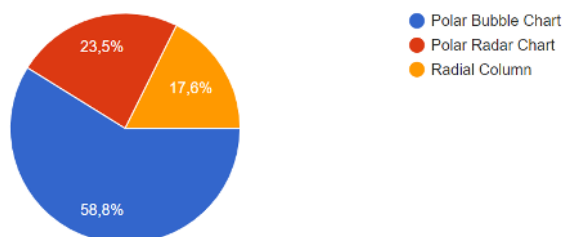


Figure 14: Mandrit Charts Evaluation

However, 23.5% of them chose the Polar Radar Chart (B) for its aesthetic that demonstrates the musical aspects in polygonal format. Even considering its chaotic complexity, they claimed that songs have a lot of information that matches the chart. And 17.6% of participants considered Radial Column (C) as more communicative and informative, according to the data collected.

During the case study for the development of Mandrit, we noticed several barriers in this area of Musical Computer Visualization, which is still maturing. These include the representational challenge of transforming musical data to make them visually communicable, accessing musical files and structured and organized information about Brazilian musical works, as well as the need for new databases to enrich the study of rhythm through digital technology.

In this same context, we consider that multidisciplinary teams as crucial to contribute to the growth and development of research in Musical Computer Visualization. The area brings together and attract artists and music professionals to explore computational capabilities and use technology as a facilitating means to support the development of tools. These tools allow to assist in the experimentation and creation of communicative processes of musical content and expand collaborative discussions between different areas.

However, we also observed that one of the greatest difficulties in this field is connecting and finding bridges with professionals in Music, due to communication bottlenecks, which generate technical dependencies.

5. Final Considerations

During this case study, we obtained data stimulating the continuity of our research in Computational Music Visualization due to the need to develop animated and interactive systems to further assist the visual communication of musical information.

Therefore, for future work, we intend to improve technical applications for automating instrumental sets that are the basis of musical arrangements and composition, and mapping individual tracks to have a greater detail of micro variations in the information of each instrument.

Also, we intend to work on treatment of musical content aided by visualizations for the recognition of patterns, similarities, and instrumental correspondences, highlighting through the graphic forms and visual structures juxtapositions, correspondences, singularities, and complexities. In addition, we aim to help in musical transcription for creating visualizations, such as animations and further studies of musical dynamics.

Seeking to expand access to musical information, we have made available in the Mandrit repository a reading base with references for researchers and future explanations in the area of MuVis. This way, we contribute to academic documentation for the construction of Computational Music Visualizations, with tutorials, database and rhythm visualizations developed in this project.

References

- [1] Hugo Carlos Cavalcanti. *Da partitura Musical: Um olhar estético a preservação da memória*. Master Dissertation, Center for Arts and Communication, Postgraduate Program in Information Science, Federal University of Pernambuco, Recife, 2013.
- [2] Delfina Malandrino, Donato Pirozzi, and Rocco Zaccagnino. Visualization and music harmony: Design, implementation, and evaluation. In *2018 22nd International Conference Information Visualisation (IV)*, pages 498–503, Fisciano, Italy, July 2018. IEEE.
- [3] Ulrich Michels. *Atlas de Musica, 1*, volume 1. Alianza Atlas, Madrid, 1987.
- [4] Ethan Hein. *Designing the Drum Loop: A constructivist iOS rhythm tutorial system for beginners*. PhD thesis, Steinhardt School of Culture, Education, and Human Development, New York University, 2013.
- [5] Giordano Cabral and Robert Willey. Analyzing Harmonic Progressions with HarmIn: the Music of Antônio Carlos Jobim. page 12, 2007.
- [6] Julian Scordato. Composing with Iannix. In *Proceedings of the Fifth Conference on Computation, Communication, Aesthetics and X.*, volume V, page 389, Lisboa, Portugal, 2017.
- [7] Tim Brown. IDEO: Design Thinking, 2018.
- [8] Maurício Vianna, Ysmar Vianna, Isabel K Adler, Brenda Lucena, and Beatriz Russo. *Design Thinking : Inovação em Negócios*. [recurso eletrônico]. MJV Press, Rio de Janeiro, 2012.
- [9] Godfried T Toussaint. *The Geometry of Musical Rhythm: What Makes a “Good” Rhythm Good?* CRC Press, 2013.
- [10] João Gomes Filho. *Gestalt do Objeto: Sistema Visual de Leitura da Forma*. Escrituras, São Paulo, 8 edition, 2008.
- [11] Fernando Luiz Vechiato and Alessandra Stefane Cândido Elias da Trindade. Encontrabilidade da informação em ambientes informacionais: diálogo teórico entre os conceitos Intencionalidade e affordance. *Prisma.com*, (42):3–20, 2020.

Fusion of Multiple Audio Representations for Music Genre Classification

Diego Furtado Silva¹, Micael Valterlânio da Silva¹,
Ricardo Szram Filho¹, Angelo Cesar Mendes da Silva²

¹Departamento de Computação – Universidade Federal de São Carlos, Brazil
Rod. Washington Luís, km 235, Jardim Guanabara – 13565-905 São Carlos, SP

²Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, Brazil
Av Trabalhador São-carlense, 400, Centro – 13566-590 São Carlos, SP

diegofs@ufscar.br, micael.sax@hotmail.com, ricardo.szram@estudante.ufscar.br, angelo.mendes@usp.br

Abstract. *Music classification is one of the most studied tasks in music information retrieval. Notably, one of the targets with high interest in this task is the music genre. In this scenario, the use of deep neural networks has led to the current state-of-the-art results. Research endeavors in this knowledge domain focus on a single feature to represent the audio in the input for the classification model. Due to this task's nature, researchers usually rely on time-frequency-based features, especially those designed to make timbre more explicit. However, the audio processing literature presents many strategies to build representations that reveal diverse characteristics of music, such as key and tempo, which may contribute with relevant information for the classification of genres. We showed an exploratory study on different neural network model fusion techniques for music genre classification with multiple features as input. Our results demonstrate that Multi-Feature Fusion Networks consistently improve the classification accuracy for suitable choices of input representations.*

1. Introduction

With the active growth of digital music distribution, mainly due to streaming platforms' success¹, organizing and retrieving musical information through computational methods has become an increasingly relevant task in both academia and industry. While the volume of data associated with music collections becomes more abundant and diverse, providing an excellent digital music experience becomes a more complex task.

Looking at improving this experience, researchers have proposed many methods for different tasks of Music Information Retrieval. Notably, machine learning has gained significant attention in this area, especially in classification tasks. The genre represents one of the most common ways of labeling music recordings. With this information, online music platforms can better organize artists and songs with similar characteristics. This organization has implications for efficient information retrieval in recommendation systems, among other tasks [1, 2, 3, 4, 5].

Most of the research on machine learning in the music domain relies on music content data, i.e., it only

considers the audio recording as the input for model learning [6, 7, 8]. Usually, research in this knowledge domain is limited to one input feature. For example, the literature on genre classification is mainly based on the use of representations that highlight timbre or rhythm characteristics [9, 10].

However, the music genre's subjectivity makes its definition quite difficult, even for humans [11]. Music from different genres may sound similar in many ways. For example, rock and blues are performed by bands with similar compositions, with the same musical instruments, and may use similar melodic constructions.

Researchers have attempted to mitigate this obstacle in recent years by using alternative data sources related to music to be categorized. One example is the multimodal classification, where different data modalities are combined, such as audio, image, and text, to further improve the result obtained [4, 12, 13, 14]. In addition, when using different input data, the results obtained are usually more accurate than those obtained from a single input [15].

Despite being a good approach and having a wide range of practical applications, different data modalities are not always available for the same song. Furthermore, improving models based on audio can also assist in the multimodal classification task's final result.

Unlike ensemble-based models, such as voting and stacking, model fusions are designed to work as an end-to-end homogeneous method to deal with multiple inputs or different transformations of the same data. Tasks that rely on multimodal learning, such as multimedia information retrieval, are good examples of fusion-based model applications [16, 17].

The two primary techniques for fusing models are early and late fusion. The main difference between them is the stage the models are fused. Early fusion techniques first extract characteristics from each input, aggregate them into a single vector, and then use a classifier to represent the combined features. In late fusion methods, each input is processed and presented to a classification model, which outputs scores for each class. Next, the outputs of those classifiers are concatenated and then used as input features to another classifier. Figure 1 illustrates the difference between these approaches.

¹IFPI issues Global Music Report 2021 -
IFPI. Available in <https://www.ifpi.org/ifpi-issues-annual-global-music-report-2021>

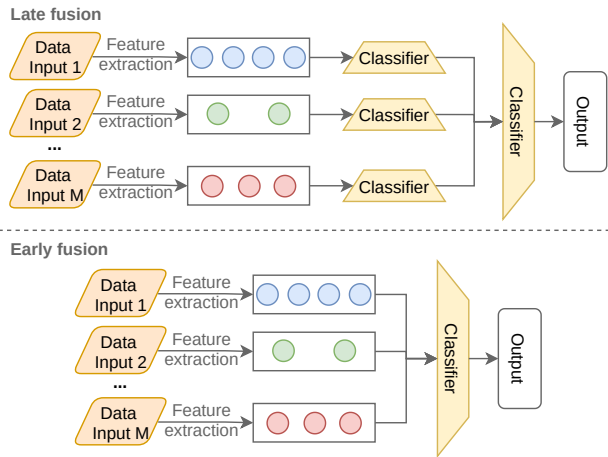


Figure 1: Illustration of late and early fusion techniques for M data inputs.

Although we can use fusion methods of machine learning techniques, deep neural networks form a convenient set of tools in this context since deep neural networks implicitly extract features based on the input data. In the case of music content, these inputs may be different audio representations of the same recording. The intuition behind using diverse representations relies on the fact that some audio processing may “hide” some interesting characteristics in a feature set that may appear in another.

In this work, we investigate ways to improve the content-based classification of music genres by combining models. For this, we use the fusion of classifiers trained on different audio representations, such as melspectrogram, constant-Q spectrum, and chroma. All models used in this work are deep learning-based networks, which means that the proposed techniques can be easily adapted to other tasks, given they extract new rich features according to the input data. Therefore, we named the general technique explored in this work as Multi-Feature Fusion Networks (MUFFN, pronounced as “muffin”).

The obtained results show that MUFFN constantly leads to more accurate results than individual models. However, some fusion models reduce accuracy, depending on the representations used and how they are combined. Therefore, this work explores different possibilities for constructing MUFFNs and presents discussions to help other researchers examine similar techniques.

2. Related Work

The literature on content-based music genre classification is considerably vast [18, 19, 7]. According to [20], the evolution of audio feature manipulation for music classification is evident when we analyze the related literature. For example, in the first efforts in this area, the genre classification was usually performed by extracting a single set of features from the whole recording and using them as input for traditional machine learning algorithms [21]. Afterward, researchers found that dividing the recordings into small frames, classify each frame individually, and provide the final answer according to the most commonly observed

label could lead us to more exciting results [22]. However, [20] approached this task using algorithms capable of aggregating frame-level features to make a global decision for the classification.

The notion that the better the feature manipulation, the better the classification results led to the application of deep learning techniques [23], which defines the state-of-the-art of music classification [9, 24]. A convolutional neural network, for instance, is capable of learning rich features from unstructured data, such as music. A usual approach uses a time-frequency-based representation of the audio as the input for a deep neural network.

However, some researchers have noted that using a single representation may not be the most appropriate way to solve some Music Information Retrieval (MIR) tasks. This observation is not limited to classification. For instance, the fusion of different features was successfully used in cover song identification [25]. On the other hand, in genre classification, some authors used traditional heterogeneous [26, 27] and homogeneous [28] ensemble techniques to accomplish the task.

Outside the task of content-based music classification, fusion models are prevalent in the multimodal classification literature [16], including music genre classification. [4] used model fusion to classify albums’ genre by their tracks (content/audio-based), its cover (image-based), and reviews on Amazon.com (text-based).

3. Multi-Feature Fusion Networks

The purpose of this work is not to create a new algorithm or neural network architecture but to discuss the multiples audio features and model relevance when used as a complementary way to discriminate a song. Despite recent work, studies like this are underexplored in the MIR community. Besides, we construct the first model for music genre classification based on this concept. Specifically, we propose Multi-Feature Fusion Networks (MUFFN) applied in music classification problems. For this, we investigate different representations and deep neural network architectures for the classification of musical genres and model fusion techniques, aiming to improve accuracy compared to individual models.

4. Experimental Setup

Since MUFFN is a general framework for creating classification models, this work’s experimental setup needs to cover various procedures. This section defines the procedures adopted to guide the search for a suitable MUFFN model for classifying musical genres. In other words, we present the datasets used, the parameters of the training phase, the decisions behind the base neural network’s choice – including a brief ablation study –, decisions on audio representations for the input, and details of the model’s fusion.

For the sake of reproducibility, we created a code repository for this work², containing source code and links

²<https://github.com/diegofurts/muffin>

to all datasets. The repository also comprises additional results and details of our proposal.

4.1. Training Procedure

We established a simple but definite training phase procedure to fairly compare models trained in a single representation and MUFFN-based models. First, we trained all the models using 5-fold cross-validation. Then, for each training fold, we separated 10% of the data for validation. We used fixed seeds for random steps in all cases, so the examples in each fold keep the same for every trained model.

We trained each model using at most 50 epochs, saving the one with the lowest loss in the validation set to test later. Besides, we applied early stopping and reducing the learning rate in plateaus to monitor the validation loss. The early stopping was set with the patience of 20 epochs (to avoid abandoning the training in the first epoch).

We defined the batch size according to memory limitations³. Specifically, we used batches of 16 examples for the two smaller datasets and only 8 for the two largest ones.

4.2. Datasets

To assess the investigated models, we used four datasets widely used in music genre recognition research: GTZAN⁴, Music Audio Benchmark Dataset (Homburg) [30], Extended Ballroom [31], and the small subset of FMA [32]. These datasets have different dimensions and a different number of labels. Table 1 summarizes the datasets.

Table 1: Summarized description of the datasets used in this work.

	Dataset			
	GTZAN	Homburg	E. Ballroom	FMA
# Recordings	1000	1886	4180	8000
# Genres	10	9	13	8
Audio duration (s)	30	10	30	30
Min. tracks/genre	100	47	23	1000
Max. tracks/genre	100	504	529	1000

We note that we applied the same experimental procedure in all datasets. Although FMA has a default train/test partition, we decided to use the same cross-validation technique for empirical consistency and better analyze the algorithms' stability in slightly different data distributions.

4.3. Audio Representations

We considered a wide range of features designed to reveal the recordings' distinct features to assess the fusion of models trained on different audio representations. However, we acknowledge that many datasets can be composed of copyright-protected songs and, therefore, only distribute

a set of features extracted from the original data [33]. Thus, we did not consider using representations used for the exact reconstruction of the original audio.

To achieve these requirements, we used the Python library librosa [34] to extract melspectrograms, mel-frequency cepstrum coefficients (MFCCs), constant-Q transformed spectrograms (CQT), tempograms, tonnetz, and the energy at the harmonics of the spectrum. Besides, we assumed that structure also performs a significant role in the music genre (e.g., pop music is usually more repetitive than jazz), so we also applied self-similarity matrices on the chromagrams.

We kept most of the librosa's default parameters for the feature extraction process. However, for the sake of dimensionality reduction and consequent better memory efficiency, we set the hop length to 1024 (the default is 512). For the same reason, we set the size of the onset auto-correlation window for the tempogram as 128. Finally, after extracting the melspectrograms, we applied a log transformation.

4.4. Deep Neural Models

Although learning algorithms based on deep neural networks present relevant results, we notice that proposing a new neural network for music genre classification is not in this work's scope. Therefore, we decided to rely on papers that presented deep learning-based methods for music classification. Initially, we experimented with models pre-trained on other domains, such as VGG16. Additionally, we considered the Convolutional Recurrent Neural Network (CRNN). However, we would need to modify the networks to deal with different representations and datasets in both cases because these architectures rely on several dimensionality reduction layers (max pooling). Thus, in some scenarios, the data dimensionality does not apply to these neural networks without losing the focus on the evaluating of different representations and fusion techniques.

Therefore, we need an architecture that is less sensitive to differences in the data dimensions. The Bottom-up Broadcast Network (BBNN) meets this requirement [35]. Besides, we experimented with CRNN, VGG16, and BBNN on the GTZAN dataset, using the melspectrogram as an input feature. BBNN achieved the best results in this experiment. Considering this, we proceed using the BBNN exclusively.

Figure 2 illustrates the BBNN. It starts with a simple convolution network, which the authors call Shallow Feature Extraction Layer (SFEL). It follows with three inception blocks, being that the output of SFEL is concatenated with each inception's output before serving as input for the next layer. These layers were named Broadcast Module (BM). Next, the authors use more convolutional and pooling layers, comprising the Transition (TL) and Decision (DL) layers. The global average pooling in the DL creates a 32-dimensional feature set used by the softmax layer to deliver a final decision. We refer the reader interested in a detailed description of the BBNN to the paper that proposes this architecture [8]. Besides, we make a

³We built our models using Keras [29], and we used Google Colab to execute our experiments because of the low resources available. For the same reason, we do not provide a runtime experiment.

⁴<http://marsyas.info/downloads/datasets.html>

better description of the network available in the paper’s repository.

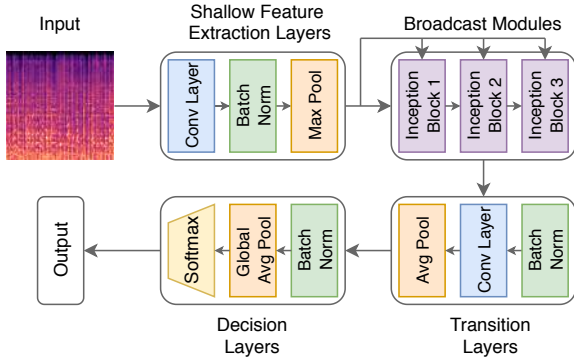


Figure 2: Simplified architecture of the Bottom-up Broadcast Network.

4.5. Ablation Study

Although finding the best neural network is not the focus of this paper, we consider it adequate to perform a short ablation study on the selected architecture. To avoid a significant overhead of this phase, we only conducted this study on GTZAN with MFCCs – given the good results on individual classifiers (c.f. Section 5) and the fact that this is a low-dimensional representation – as input. Also, we used the 60-20-20 split for training, validation, and test sets. Note that it implies a slightly smaller training set than the other experiments. Therefore, the training phase is marginally faster, and we can better understand the effects on validation and test.

For this, we realized experiments using one, two, and all three inception blocks of BBNN. Besides, we note that these blocks have four parallel sequences of layers (c.f. [8] for details). Therefore, we experimented with the network with all inception blocks but removed one of these sequences, composing four new networks.

We note that the BBNN is notably prone to overfitting in all experiments performed in this paper. Achieving perfect accuracy in the first 20 training epochs is common. On the other hand, the model achieves much lower results in the validation and test sets. However, the loss function usually keeps decreasing in both training and validation.

In all ablation experiments, this effect was reduced. As a result, the epoch that achieves the perfect score for the training set occurs later. However, it does not reflect on better generalization and, consequently, better accuracy. In all cases, the complete network achieves the best result.

For example, while the complete network’s accuracy is 0.799, removing one or two inception blocks leads to 0.784 and 0.776, respectively. In addition, the accuracy obtained when we removed one of the inception paths varied from 0.773 to 0.793. Although these differences are not significant, we decided to keep using the complete architecture.

4.6. Early and Late Fusion

Once we have defined the neural network to use, we need to define which points to concatenate to build a single model. In the case of late fusion, each input must pass through its entire BBNN. The output of each input’s softmax layer is concatenated and then used as an input for another softmax layer for the final decision. Figure 3 illustrates this approach.

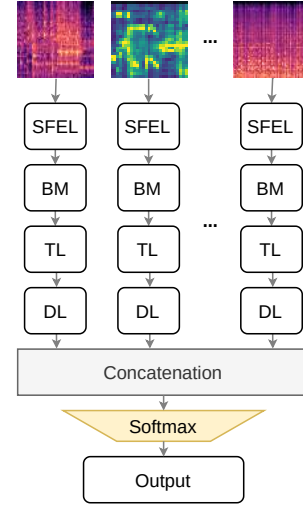


Figure 3: Late fusion scheme when using BBNN-based MUFFN.

We must define a layer that represents the features extracted by the neural network for early fusion models. These features can be obtained after any set of layers illustrated in Figure 2 or after each inception block. However, as shown in the ablation study, we noticed that using all layers of BBNN leads to better results. Therefore, we use the features aggregated by the global average pooling layer. Figure 4 illustrates the approach to building models using early fusion.

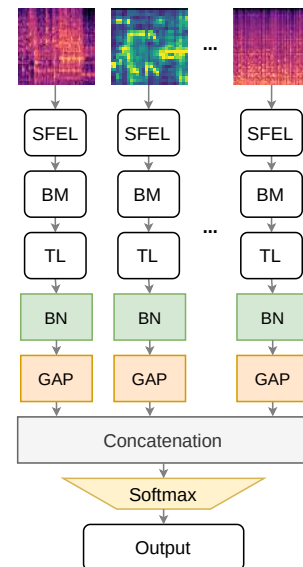


Figure 4: Early fusion scheme when using BBNN-based MUFFN.

It is important to note that we first train individual models and store their weights to train the fusion models. From that, we use these pre-trained models to build MUFFN models. We then evaluate data fusion methods with fixed weights and with fine-tuning the weights through model retraining. In the latter case, we only allow the training procedure to adjust the weights of the transition and decision layers.

5. Results

This section presents the classification results, following the experimental setup introduced in Section 4.

5.1. Individual Classifiers

We started the presentation of results by the individual classifiers. We note that, as mentioned in Section 4.3, we discarded some of the representations considered in this work for preliminary experiments. For example, the accuracy obtained using tonnetz was 0.113 in the GTZAN dataset. On the other hand, the use of self-similarity matrices led to 0.310 of accuracy in the Homburg dataset, probably due to the recordings' short duration.

Table 2 shows the results obtained by using different representations in all datasets.

Table 2: Accuracy and standard deviation obtained by individual classifiers.

Feature	GTZAN	Homburg	E. Ballroom	FMA
Chroma	0.696±0.046	0.536±0.043	0.878±0.012	0.483±0.039
CQT	0.785±0.013	0.639±0.033	0.932±0.006	0.643±0.010
MFCC	0.774±0.037	0.585±0.029	0.926±0.013	0.619±0.024
Melspec	0.851±0.022	0.644±0.031	0.941±0.006	0.654±0.016
Tempo	0.478±0.028	0.453±0.026	0.886±0.088	0.432±0.011

We note that the melspectrogram was the best representation in all datasets. In some cases, especially in the GTZAN dataset, this difference is substantial compared to the second-best result. This observation is aligned with the most common assumption in the literature that this is the best representation for the content-based music classification task. Moreover, we observe that using the constant-Q transform consistently leads to the second-best results, and the MFCC achieves accuracy rates close to those obtained by CQT.

5.2. Fusion Models

In this section, we gradually built the base to arrive at the final MUFFN models. For every decision, we first evaluate if it seems promising in the smaller datasets, i.e., GTZAN and Homburg. Then, if it presented good results, we replicated the decision to the other datasets.

We started our decision by assuming that the melspectrogram should be part of all assessed subsets of representations. Then, we used this assumption to create two double-input fusion networks: melspectrogram and MFCC; and melspectrogram and chroma. In the first case, we evaluate if two representations based on the same

premise of applying the mel scale improve timbre representation. In the latter case, we assess if tonality- and timbre-related features may complement each other. Besides, we added the tempogram as a third input to the second approach, creating a model that fuses tonality, timbre, and tempo features. We used these three combinations in the first batch of experiments.

We note that we evaluated early and late fusion schemes and training with and without fine-tuning, as described in Section 4.6. In all cases being assessed, early fusion methods achieved better results than late fusion. This phenomenon is more evident when we allow fine-tuning. In some cases, fine-tuning decreases the accuracy of late fusion models. On the other hand, it constantly improves the results of early fused networks. For this reason, we continue our analysis considering only early-fused and fine-tuned models.

Figure 5 illustrates this fact in the specific case of GTZAN with MFCC and melspectrogram as inputs.

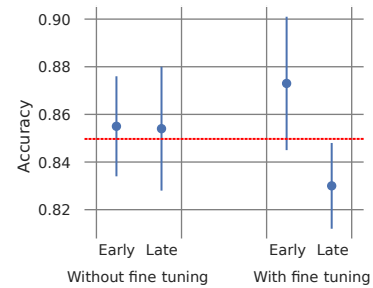


Figure 5: Comparison between early and late fusion, with and without fine tuning, of melspectrogram and MFCC in the GTZAN dataset. The red dashed line shows the accuracy of the best individual model.

Note that the figure results are mostly better than the individual classifier, which also happens on Homburg data. For this reason, we also evaluated this fusion in the remaining datasets. However, the same observations do not apply to the other described fusion. The only exception is the fusion of melspectrogram, chroma features, and tempogram with Homburg data. In this case, the MUFFN achieved 0.002 of accuracy improvement. On the other hand, on GTZAN, the same architecture caused an accuracy reduction of 0.023.

We realized that merging the CQT spectrum caused improved results from the successful merger in the previously described step. Therefore, for this new merger, we evaluated the effects of adding the chromagram or the tempogram. Finally, we evaluate adding these last two representations simultaneously, creating a fusion model of five audio representations.

For a better presentation and interpretation, Table 3 summarizes these fusions and attributes short identifiers to each of them.

Figure 6 shows the accuracy obtained for these models and the best individual one, obtained with melspec-

Table 3: Subsets of audio representations used to create MUFFN models.

Identification	Subset of input representations
MM	Melspectrogram + MFCC
MMC	MM + CQT spectrum
MMCC	MMC + chromagram
MMCT	MMC + tempogram
MMCCT	MMC + chromagram + tempogram

trogram.

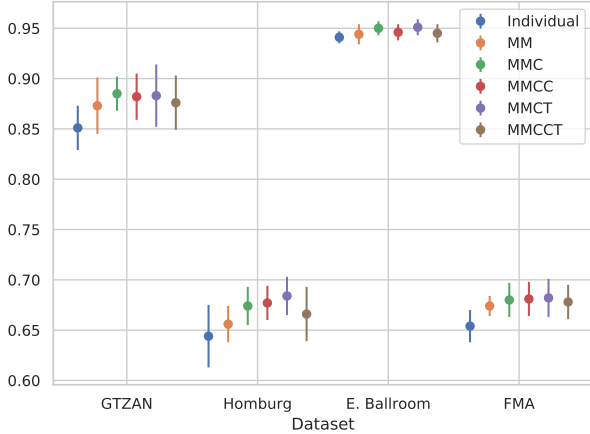


Figure 6: Classification results using individual and fusion models.

For further interpretation and analysis, Table 4 presents the same results, but numerically.

Table 4: Accuracy and standard deviation obtained by the best individual and fusion-based classifiers.

Model	GTZAN	Homburg	E. Ballroom	FMA
Individual	0.851±0.022	0.644±0.031	0.941±0.006	0.654±0.016
MM	0.873±0.028	0.656±0.018	0.944±0.010	0.674±0.010
MMC	0.885±0.017	0.674±0.019	0.950±0.007	0.680±0.017
MMCC	0.882±0.023	0.677±0.017	0.946±0.008	0.681±0.017
MMCT	0.883±0.031	0.684±0.019	0.951±0.008	0.682±0.019*
MMCCT	0.876±0.027	0.666±0.027	0.945±0.009	0.678±0.017

We note that, in general, adding other representations can improve the results. In contrast, the two representations that add chroma features achieved lower accuracy in most cases than the respective version without such a representation. It is easily observed by comparing MMC against MMCC and, especially, MMCT against MMCCT.

6. Discussion

The results presented show that there is great potential for improving content-based music classification. We recall exploring different neural network architectures with a single representation to find a robust model in this scenario. It means that our experiments showed that the use of MUFFN could improve the classification even compared to a powerful competitor.

Unfortunately, that gain comes at a cost. Aggregating different neural networks increases the number of parameters to be trained, generating a higher computational cost. An advantage in this sense that we observed in our experiments is that we achieved good results without training networks from scratch. The fine-tuning strategy used does not extend across the entire network but only to a reduced set of final layers of each model.

The number of parameters that BBNN needs to train for a single-representation model is 185,642. Besides, it has 5,184 non-trainable parameters. For example, the number of parameters in MUFFN with four distinct inputs and an early fusion approach is 763,274. However, only 54,922 of them are trainable. Therefore, if we intended to train the model from scratch, we would need to adjust 742,538 of these parameters.

One way to reduce overhead for the design and training of a MUFFN is to choose the appropriate inputs. We have not yet carried out an in-depth study in this regard due to its scope. However, it was possible to observe some patterns. For example, when adding chromagram as one of the inputs, the accuracy tendency is to maintain or decrease. For instance, the accuracy obtained by MMCCT is consistently lower than MMCT. However, the same does not happen with the insertion of tempograms. This aspect probably appears because the tempogram adds complementary information to representations based on the mel scale, which does not occur for chroma features.

With these results, we believe that using MUFFN can significantly improve the results of specific tasks. For example, consider the task of hierarchical classification. It is intuitive to think that representations associated with timbre are sufficient at a higher hierarchy level. The instruments used in classical and popular music tend to be quite different, for example, in a melspectrogram. However, at a level that we need to differentiate between genres that use similar instruments and techniques, other representations can be used to “disambiguate” decisions.

For these reasons, we believe that the obtained results are motivating for future work in this direction.

7. Final Remarks

As we pointed out a few times during this work, our work presents a general framework that can be modified in several aspects. Furthermore, we demonstrate that MUFFN models can improve music genre classification accuracy in all assessed datasets with a few options evaluated. Therefore, we hope that our work will seed other research endeavors to create models based on MUFFN.

We note that we have not studied the contribution of each input in this work. Thus, despite discussing some clues, we cannot provide clear interpretations to guide the choice of representation inputs. Therefore, we intend to conduct an in-depth study to understand this aspect better and then provide more precise guidelines for the design of MUFFN for MIR tasks.

Furthermore, we acknowledge that neural networks may automatically learn features from the melspectrogram, such as MFCC or CQT, for a sufficiently large dataset. However, it depends on many factors, including the network architecture. As MUFFN is a framework, we intend to evaluate how these observations stand in different scenarios.

As a future direction, we intend to evaluate other neural network architectures as the basis for the fusion model. We limited ourselves to only one architecture in this work to avoid missing the central proposal's focus. However, we believe that different architectures can achieve better results for each diverse input representation and, consequently, for the fused model. Furthermore, as we acknowledge our proposal adds complexity to the models, we intend to focus on more efficient architectures and training strategies.

Finally, we intend to use MUFFN in other MIR tasks, such as tagging, emotion recognition, content-based similarity, and genre identification in multi-label and hierarchical classification scenarios.

8. Acknowledgments

This work was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code PROEX-12049601/D, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – Finance Code 2019/08653-0.

References

- [1] Paul Lamere. Social tagging and music information retrieval. *Journal of new music research*, 37(2):101–114, 2008.
- [2] Peter Knees and Markus Schedl. Music retrieval and recommendation: A tutorial overview. In *International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1133–1136. ACM, 2015.
- [3] Yan Wan. *Deep Learning for Music Classification*. PhD thesis, Hong Kong University of Science and Technology, 2016.
- [4] Sergio Oramas, Francesco Barbieri, Oriol Nieto, and Xavier Serra. Multimodal deep learning for music genre classification. *Transactions of the International Society for Music Information Retrieval*, 1(1):4–21, 2018.
- [5] Sandy Manolios, Alan Hanjalic, and Cynthia CS Liem. The influence of personal values on music taste: towards value-based music recommendations. In *ACM Conference on Recommender Systems*, pages 501–505, 2019.
- [6] Markus Schedl, Emilia Gómez, Julián Urbano, et al. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8(2-3):127–261, 2014.
- [7] YV Murthy and Shashidhar G Koolagudi. Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. *ACM Computing Surveys*, 51(3):45, 2018.
- [8] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. Bottom-up broadcast neural network for music genre classification. *arXiv preprint arXiv:1901.08928*, 2019.
- [9] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017.
- [10] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.
- [11] Klaus Seyerlehner. *Content-based music recommender systems: Beyond simple frame-level audio similarity*. PhD thesis, Johannes Kepler Universität Linz, 2010.
- [12] Federico Simonetta, Stavros Ntalampiras, and Federico Avanzini. Multimodal music information processing and retrieval: Survey and future challenges. In *2019 International Workshop on Multilayer Music Representation and Processing (MMRP)*, pages 10–18. IEEE, 2019.
- [13] W. Guo, J. Wang, and S. Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.
- [14] Chao Zhang, Zichao Yang, Xiaodong He, and Li Deng. Multimodal intelligence: Representation learning, information fusion, and applications. *IEEE Journal of Selected Topics in Signal Processing*, 14(3):478–493, Mar 2020.
- [15] Jaehun Kim, Julián Urbano, Cynthia C. S. Liem, and Alan Hanjalic. One deep music representation to rule them all? a comparative analysis of different representation learning strategies. *Neural Computing and Applications*, 32:1067–1093, 2020.
- [16] Dhanesh Ramachandram and Graham W Taylor. Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6):96–108, 2017.
- [17] Wenwu Zhu, Xin Wang, and Hongzhi Li. Multi-modal deep analysis for multimedia. *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.
- [18] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2010.
- [19] Bob L Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *Journal of New Music Research*, 43(2):147–172, 2014.
- [20] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and adaboost for music classification. *Machine Learning*, 65(2-3):473–484, 2006.
- [21] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.
- [22] Kris West and Stephen Cox. Finding an optimal segmentation for audio genre classification. In *International Society for Music Information Retrieval Conference*, pages 680–685, 2005.
- [23] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6959–6963. IEEE, 2014.
- [24] Jaehun Kim, Julián Urbano, Cynthia CS Liem, and Alan Hanjalic. One deep music representation to rule them all? a comparative analysis of different representation learning strategies. *Neural Computing and Applications*, 32(4):1067–1093, 2020.

- [25] Christopher J Tralie. Early MFCC and HPCP fusion for robust cover song identification. In *International Society for Music Information Retrieval Conference*, pages 294–301, 2017.
- [26] Loris Nanni, Yandre MG Costa, Rafael L Aguiar, Carlos N Silla Jr, and Sheryl Brahnam. Ensemble of deep learning, visual and acoustic features for music genre classification. *Journal of New Music Research*, 47(4):383–397, 2018.
- [27] Rodolfo M Pereira, Yandre MG Costa, Rafael L Aguiar, Alceu S Britto, Luiz ES Oliveira, and Carlos N Silla. Representation learning vs. handcrafted features for music genre classification. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.
- [28] Deepanway Ghosal and Maheshkumar H Kolekar. Music genre recognition using deep neural networks and transfer learning. In *Interspeech*, volume 2018, pages 2087–2091, 2018.
- [29] Francois Chollet. *Deep Learning mit Python und Keras: Das Praxis-Handbuch vom Entwickler der Keras-Bibliothek*. MITP-Verlags GmbH & Co. KG, 2018.
- [30] Helge Homburg, Ingo Mierswa, Bulent Moller, Katharina Morik, and Michael Wurst. A benchmark dataset for audio classification and clustering. In *International Society for Music Information Retrieval Conference*, pages 528–531, 2005.
- [31] Ugo Marchand and Geoffroy Peeters. The extended ballroom dataset, 2016.
- [32] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. In *International Society for Music Information Retrieval Conference*, pages 316–323, 2017.
- [33] Wenqin Chen, Jessica Keast, Jordan Moody, Corinne Moriarty, Felicia Villalobos, Virtue Winter, Xueqi Zhang, Xu-anqi Lyu, Elizabeth Freeman, Jessie Wang, Sherry Kai, and Katherine M. Kinnaird. Data usage in mir: History & future recommendations. In *International Society for Music Information Retrieval Conference*, pages 25–32, 2019.
- [34] 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*, volume 8, pages 18–25, 2015.
- [35] Caifeng Liu, Lin Feng, Guochao Liu, Huibing Wang, and Shenglan Liu. Bottom-up broadcast neural network for music genre classification. *Multim. Tools Appl.*, 80:7313–7331, 2021.

Towards Real-time Score Analysis in PWGL

Mika Kuuskankare¹

¹Independent grant researcher

mkuuskan@yahoo.com

Abstract. *In this paper, we introduce an original approach to computerized music analysis within the graphical computer-assisted composition environment called PWGL. Our aim is to facilitate real-time analysis of interactive scores written in common Western music notation. To this end, we have developed a novel library that allows us to analyze scores realized with the help of ENP (the graphical music notation module of PWGL), and to visualize the results of the analysis in real-time. ENP is extended to support the display of supplementary information that can be drawn on top of the score as an overlay. The analysis backend is realized with the help of our built-in musical scripting language based on pattern matching. The analysis results are presented directly as a part of the original score leveraging the extensible and interactive visualization capabilities of ENP.*

In this paper we describe the current state of the library and present, as a case study, a fully functional application, which allows for the real-time analysis and display of voice leading errors according to the counterpoint rules developed mainly in the Renaissance and Baroque eras.

1. Introduction

Typically, computational musicology employs *non-real-time* algorithms to analyze music encoded in textual form. Within the computer music community, *real-time music analysis* has become somewhat synonymous to real-time audio analysis and visualization. In this paper, we focus on *analysis of interactive scores written in common Western music notation*, an approach still underrepresented within the community mainly because of the complex nature of computerized music notation.

It is customary to divide real-time systems into two main categories: hard real-time and soft real-time. Systems falling under the first category usually present either safety or mission critical constraints and, in general, missing a deadline is considered disastrous. Systems in the second category work "fast enough" and missed deadlines affect only the quality of service. This article won't debate the fine points of real-time computing. Instead, for the purposes of this paper, we consider the latter category, specifically in the sense of fast-or fast enough-computation.

Encapsulating the notational and analytical practices of the western art music in a real-time context, requires versatile and interactive representations of music, flexible visualization devices, and musical knowledge encoded into the system. Let us consider the representation of music in state-of-the-art real-time and non-real-time musical environments. *Real-time* environments, such as Pure Data[1], concentrate on producing and manipulating sound, video, and graphics, rarely providing robust

interactive representations of musical notation because of the computational complexity associated with these operations. *Non-real-time* environments, such as OpenMusic[2] for computer-assisted composition or LilyPond[3] for music notation, allow for robust musical representations at the expense of interactivity.

Rich music analytical representations and real-time interactivity need not to be mutually exclusive, instead, the two should complement each other. Text editors, for example, allow people to develop documents in real-time while retaining a rich view of these documents. The same is true of photo editing, where users can apply sophisticated filters to their photos and instantly see the results.

PWGL[4] can already be used to analyze musical scores realized with the help ENP[5] using a variety of methods[6, 7, 8, 9]. However, the existing methods have not yet been adapted for real-time analysis. In this paper, we present a novel PWGL library, that aims to make it possible to analyze ENP scores and to visualize the results of the analysis in real-time. The analysis backend is realized using ENP-Script[10].

PWGL is a visual music programming language written in Lisp and OpenGL. Its primary focus is on computer-assisted composition in an integrated environment with music notation and software synthesis. ENP, in turn, is PWGL's built-in music notation program developed primarily for applications concerning computer-assisted composition and virtual instrument control. It is designed to produce automatic, reasonable quality musical typesetting. Finally, ENP-Script is the native scripting language of ENP. It allows us to access complex musical patterns with the help of a pattern-matching syntax developed for our constraint-based compositional system.

The rest of the paper is organized as follows. First, we introduce the two key software components of the library: ENP and ENP-Script. Next, we discuss implementation details and present an application prototype that allows for real-time analysis and display of common voice leading errors. The paper ends with some concluding remarks and ideas for future work.

2. ENP

ENP is a full-featured music notation program with a graphical user-interface and an extensive set of graphical devices for enriching notational output. ENP can produce relatively complex, automatically typeset and interactive music notation (see Figure 3).

At the center of the expressive power of ENP are interactive and user-definable graphical devices, called

expressions. In ENP, the term expression is used in a broader sense than as a traditional expression marking. ENP-Expressions are multipurpose visualization devices that can be used to represent complex dynamic Lisp-based objects as a part of a musical texture (see Figure 4). In addition to their traditional use, they can be used in a wide range of applications, such as providing control information for virtual instrument synthesis. One of the unique features of ENP-Expressions is that they can be attached to a discontinuous group of objects spanning across multiple parts. This is especially beneficial in musical analysis applications that typically deal with the relationships between independent melodic lines and simultaneously sounding harmonies.

3. ENP-Script

ENP-Script is the scripting language of ENP. It uses the syntax of the pattern-matching language of PWConstraints[11]. PWConstraints, in turn, is a compositional system that uses the Backtracking algorithm to obtain a solution to musical constraint satisfaction problems. ENP-Script, while sharing most of its behavior with PWConstraints (e.g., language syntax, score access), does not use backtracking. The scripts consist of one or more scripting rules. The scripting rule, in turn, consists of a pattern-matching part and a Lisp-code part. The pattern-matching part contains a set of pattern-matching variables each representing an object in the score. Pattern-matching variables hold information about the pitch of the associated notes, rhythm, current harmony, etc. Once the pattern-matching part of a rule has accessed the desired objects (i.e., notes, chords, measures, harmonic formations), these variables can be used in the Lisp-code part. To access information from the variables, the 'm' method is typically used ('m' standing for multi-accessor). Normally, the m method returns either a single note for non-compound objects or a list of notes for compound objects (e.g., chords, beats, measures, harmonic formations). It also accepts a list of keyword arguments that allow more precise specification of the type of data that will be returned. Finally, the scripting engine loops through the score's notational objects and applies the scripting rules, usually to produce some effect, such as attaching an expression. For more detailed discussion of ENP-Script, refer to reference [10].

4. The Library

The present library contains two modules: (1) the *visualization module*, and (2) the *analysis module*. The analyze-visualize cycle is triggered every time a property of the score is changed. This typically happens when the user, for example, transposes a note.

4.1. The Visualization Module

The results of the analysis are visualized directly on the source score. For the purposes of the present library, a new category of ENP-Expressions—called *overlay-expression*—was developed. Overlay-expressions differ from standard ones in a few specific ways: (1) they are not part of the

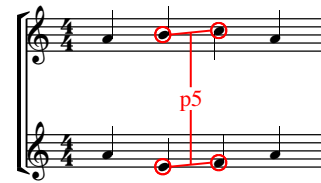


Figure 1: The voice-leading expression indicating the notes involved as well as the name of the error ('P5' indicates a parallel).

object structure of the score, i.e., they are not saved with the associated score, and (2) they are temporary, i.e., the lifetime of the overlay objects ends before the next batch of objects begins to be initialized. Instead, as the name suggests, they are drawn as an overlay on top of existing score objects. Most expressions, such as dynamics or articulations, are usually attached to objects in one single part, therefore relying on the local coordinate system of that part. However, an overlay-expression refers to the global positions of the objects it is associated with and, therefore, can easily extend itself across different parts (as can be seen in Figure 1).

The library implements two expressions that extend the overlay-expression. These are called the *voice-leading expression* and the *voicing expression*. Figure 1 shows the voice-leading expression. It holds a reference to a pair of notes in two different parts. Visually, it connects the notes in each part with a line that has a hollow circle on each end. A vertical line connects the note pairs involved in the offending voice leading case and displays an error code (e.g., 'P5' for parallel fifths) around the midpoint of the line segment. The voicing-expression, in turn, marks cases in the score where there is either a voice crossing, or an open spacing. The voicing-expression holds a reference to the corresponding notes that—by definition—have to be in two different parts. The two notes are encircled and connected vertically with a dotted line. The name of the error is again indicated around the midpoint of the line segment.

4.2. The Analysis Module

The analysis module is implemented in two parts: (1) as a real-time analysis engine, and (2) as a set of analysis rules.

The real-time analysis engine implements a slightly modified version of our scripting engine. One of the major differences is that the additional structures (see Figure 2) prepared in advance to assist in accessing the multitude of different score objects (notes, harmonies, etc.) are cached instead of being regenerated every time a script is run. One of the most complex of these structures is the harmonic structure shown in Figure 2. As a further optimization, the scripts are also only applied to the part of the score that is visible to the user.

The complete set of rules implemented by the library currently contains more than 10 rules. For the purposes of this paper, we consider only the following

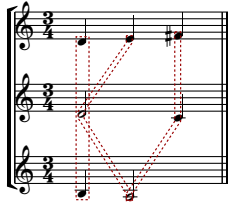


Figure 2: To assist and optimize the search process, several auxiliary structures are generated for both our constraints and scripting engines. The simultaneity (i.e., harmonic) structure is shown here. The polygons enclose the three possible simultaneities in this particular score.

rules: (1) parallel fifths and octaves, (2) voice crossing, and (3) open spacing. Parallel fifths and octaves are progressions in which the interval of a perfect fifth or an octave is followed by the same interval between the same two parts. Voice crossing is the intersection of melodic lines, e.g., a lower voice crosses above a higher one. Spacing, refers to the simultaneous vertical placement of notes in relation to each other. Open spacing typically occurs when the interval between consecutive voices exceeds an octave. Note, that the details about the interpretation of different rules typically varies from period to period.

Next, we will examine in detail the code for detecting parallel fifths progressions shown in Listing 1. Line 1 of Listing 1, defines the pattern and type of score objects (indicated by the keyword `:harmony`) to which the rule is applied. The pattern consists of a star (`'*'`) which matches zero or more score objects and a variable name (`'?1'`). This particular pattern extracts all the harmonic formations (groups of simultaneously sounding notes) from the score and binds them one by one to the variable named `'?1'`. In line 3, we call the `m-method` with the keyword `':vl-matrix'` as an argument to return an object of type `vl-matrix` (voice leading matrix). The `vl-matrix` object encapsulates the relevant notes that participate in all the possible voice leading cases between consecutive simultaneities. In four part texture this results in at most six different cases: voice 1 against voices 2, 3, and 4; voice 2 against voices 3 and 4; and voice 3 against voice 4. The presence of a rest in any of the voices naturally reduces the number of combinations. In line 5, we access the `vl-matrix` object to retrieve a list of notes belonging to the top-most voice. Next, we check that the voice actually moves up or down, otherwise this voice will not contribute to any parallel movement. Next, we access the pitch properties of the two notes and check if they are equal (line 6). In case the pitches are not equal, we need to loop through all the remaining voices to check for parallel fifths (line 7). For each voice, we again make sure that it is moving either up or down (line 9). If so, we proceed with the remaining checks. Lines 10-11 check that intervals between the two voices are both perfect fifths, while lines 12-13, in turn, check that both voices also move in the *same* direction.

If all the aforementioned conditions are satisfied,

the two voices move in parallel fifths. To indicate this in the score, we instantiate an overlay-expression that is subsequently attached to the formation of the four notes (two notes in each of the involved parts). This is done in lines 14-17. The overlay-expressions have many user-definable properties, one of which is color. Here, we chose to display the overlay in red (see line 17) to denote a high level of urgency. The actual drawing of the overlays is delayed until all the voice leading rules have been checked.

5. The Application

As a proof of concept, we have built a fully functional application that allows for the analysis and display of common voice-leading errors. The application is shown in Figure 5. The current set of available rules is displayed on the right as checkboxes. These GUI components can be used to activate or de-activate rules. Only errors selected by the user are checked. The related visualizations are updated in real-time as the user works on the score. In addition to pitch, the user can manipulate any other score property, including rhythm.

6. Performance Measures

We collected some preliminary performance measures with the help of our application prototype. The tests were performed on a 2012 Apple MacBook Pro with an Intel Core i7 processor running at 2.7 GHz. The results are presented in Table 1. The 'total' time includes the initial layout calculation and drawing of the score (≈ 10 ms), the analysis phase (≈ 15 ms) and the drawing of any overlay objects (≈ 2 ms). The response time of slightly under 30 ms seems to be in the realm of real-time and the GUI response confirms this. Thus, our experiment aligns well with our initial objectives.

Step	Time (ms)
layout/draw	10
analyze	15 ¹
visualize	2
total	27

Table 1: The approximate timings of the steps involved in the analysis process.

¹ ≈ 35 ms without caching

7. Acknowledgements

This work has been supported by the Jenny and Antti Wihuri Foundation.

8. Conclusions

In this paper we presented a novel PWGL library that allows for the real-time analysis of musical scores written in the ENP score format. The potential application fields range from elementary music pedagogy to advanced counterpoint exercises, as well as computer-assisted composition. Our work could be applied, for example, in computer-based training, making it possible to develop interactive

Listing 1: Code for detecting parallel fifth progressions in an ENP score

```

1 (* ?1 :harmony
2   (?if
3     (let ((mat (matrix-access (m ?1 :vl-matrix 2 :object t) :h)))
4       (when mat
5         (destructuring-bind (upper-1 upper-2) (first mat)
6           (unless (= (m upper-1) (m upper-2))
7             (loop for mel2 in (rest mat) do
8               (destructuring-bind (lower-1 lower-2) mel2
9                 (unless (/= (m lower-1) (m lower-2))
10                  (when (and (= (mod12 (abs (- (m upper-1) (m lower-1)))) 7)
11                    (= (mod12 (abs (- (m upper-2) (m lower-2)))) 7)
12                    (= (signum (- (m upper-2) (m upper-1)))
13                      (signum (- (m lower-2) (m lower-1))))))
14                  (add-overlay upper-1 upper-2 lower-1 lower-2
15                             :kind :voice-leading
16                             :id :parallel-fifth
17                             :color :red))))))))))

```

music theory applications. One possible application could be an advanced counterpoint tutor with built-in knowledge of common voice-leading and harmony rules. It would display the assignments in music notation, and correct and instruct students in real-time using a combination of text, colors, and commonly accepted music theoretical markings. Being able to see the analysis in real-time can be an effective way to gain understanding of how different elements of the composition (e.g., harmony, melody) work together. Furthermore, our tool could be used not only as a music analysis tool, but also as a compositional tool. Different kinds of rules, such as those keeping track of harmonic or melodic pitch class sets, could be used as visual reminders. Additionally, more sophisticated rules (e.g., checking the correctness of Harp pedaling) could be used to assist in the compositional process to allow the composer to focus on other tasks. Finally, the user could also invent their own expressions, based on the overlay-expression, to fit their particular needs.

Currently, the library is under active development. There will undoubtedly be ample opportunities for performance optimizations, such as precompiling and caching the analysis rules themselves. We are also working on additional analysis backends, one of which will be based on the Humdrum Toolkit[12]. The Humdrum Toolkit is a widely-used open-source software package for musicological research developed at CCARH at Stanford University. It would allow us to access the rich and readily available music analysis features of Humdrum and a vast number of freely available pieces encoded in the kern data format. The different backends would not be mutually exclusive; instead, they would complement each other.

References

- [1] Miller Puckette. Using Pd as a score language. In *Proceedings of International Computer Music Conference*, pages 184–187, 2002.
- [2] Jean Bresson and Carlos Agon. Visual programming and music score generation with openmusic. In *IEEE Symposium on Visual Languages and Human-Centric Computing*, 2011.
- [3] Han-Wen Nienhuys and Jan Nieuwenhuizen. LilyPond, a system for automated music engraving. In *XIV Colloquium on Musical Informatics (XIV CIM 2003)*, Firenze, Italy, 2003.
- [4] Mikael Laurson, Mika Kuuskankare, and Vesa Norilo. An Overview of PWGL, a Visual Programming Environment for Music. *Computer Music Journal*, 33(1):19–31, 2009.
- [5] Mika Kuuskankare and Mikael Laurson. Expressive Notation Package. *Computer Music Journal*, 30(4):67–79, 2006.
- [6] Mikael Laurson, Mika Kuuskankare, and Kimmo Kuitunen. Introduction to computer-assisted music analysis in PWGL. In *Sound and Music Computing '05*, 2005.
- [7] Mikael Laurson, Mika Kuuskankare, and Kimmo Kuitunen. The Visualisation of Computer-assisted Music Analysis Information in PWGL. *Journal of New Music Research*, 37(1):61–76, 2008.
- [8] Mika Kuuskankare. Schenkerian analysis tools in ENP. In *Proceedings International Computer Music Conference*, 2013.
- [9] Mika Kuuskankare and Craig Sapp. Visual Humdrum-library for PWGL. In *Proceedings of ISMIR*, 2013.
- [10] Mika Kuuskankare and Mikael Laurson. Intelligent Scripting in ENP using PWConstraints. In *Proceedings of International Computer Music Conference*, pages 684–687, Miami, USA, 2004.
- [11] Mikael Laurson. PWConstraints. In G. Haus and I. Pighi, editors, *X Colloquio di Informatica Musicale*, pages 332–335, Milano, 1993. Associazione di Informatica Musicale Italiana.
- [12] David Huron. Music information processing using the Humdrum Toolkit: Concepts, examples, and lessons. *Computer Music Journal*, 26(2):15–30, 2002.

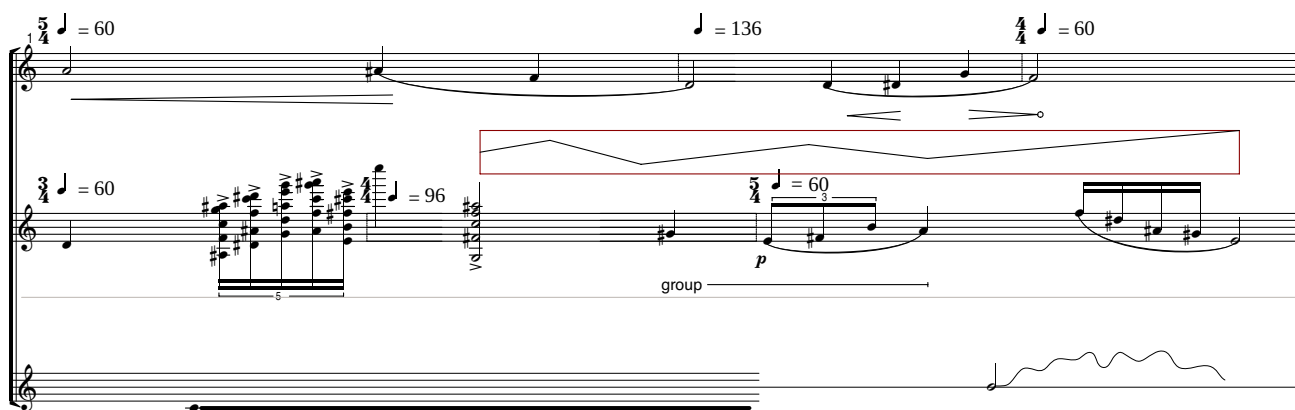


Figure 3: A relatively complex score notated with the help of ENP. Besides standard notational devices, such as articulations and dynamics, the score showcases several types of graphical control information as well as different simultaneous tempi and time signatures.

Figure 4: An example of different kinds of analytical markings. Here, we use a combination of text and colored areas to highlight motives as well as commonly accepted music theoretical markings, such as set class names to display the total harmonic content of selected simultaneities.

Figure 5: The application displaying common voice-leading errors.

Evaluating the Automatic Chord Estimation and Alignments tasks needs using metrics from a code challenge

Valter Jorge da Silva¹, Giordano Ribeiro Eulalio Cabral¹

¹Centro de Informática – Universidade Federal de Pernambuco
Av. Jorn. Aníbal Fernandes, s/n, – 50740-560 Recife, PE

vjs3@cin.ufpe.br, grec@cin.ufpe.br

Abstract. *Automatic Chord Estimation is a subject of Music Information Retrieval that tries to extract the chords of a song in a usable manner. In the last year, many researchers tried to overperform the quantitative metrics, but the results lack reproducibility by who needs them, musicians. In this article, we reviewed the state of the art of some of these areas and performed a code Challenge that was evaluated by some of the MIREX metrics and by musicians. Then, with these results, we evaluated the need for evolution on the Estimation task and the Alignment Task of the MIR area.*

1. Introduction

Within the Music Information Retrieval (MIR) process, various methods of extracting information from songs are available. One of the most discussed topics is chord identification in an automated way, with is called Automatic Chord Estimation (ACE) [1, 2]. In ACE it is proposed that a system can identify which chord is being played at a moment, given an input that can be a song or just an independent sound signal. The use of ACE could improve several aspects of the music area, such as teaching, accompaniment activities (tag-along), meta-information extraction (genre, duration, etc), automatic creation of chords sheets or scores, among others.

This area has been receiving innovative methods in recent years, but it is going through a period of stagnation. Even with new technologies emerging, such as the use of Deep Learning, the tools are created focusing on quantitative results for the problem, such as general accuracy, or how much more it estimate right. This situation creates a problem in which applications focus on making fewer mistakes, evolving in small percentage portions of the accuracy metric between one tool or another, but causing a gap in the qualitative results.

The qualitative results of a system with this proposal must be related to how much the output can be used for the final purpose of the area, being this use given by an automatic tool or by a musician. For example, a result of these systems can hit 90% of all samples, but the 10% that it misses results in a final result that is a non-reproducible input, as it is very segmented.

Another problem related to the quality of the results is the issue of aligning the chords with the original audio. In ACE tools, the output usually has a timestamp telling you when the chord change happened. However, these resulting times usually have a temporal discrepancy

in relation to the original audio, necessitating the use of another technique called Chord Alignment.

Automatic Chord Alignment (ACA) is a topic in the MIR area in which techniques for performing the temporal alignment of chords are proposed [4, 5]. Given the audio of a song and the chords played in it, a Chord Alignment system is capable of synchronizing what is in the written language with the audio it represents.

In this article, a review was carried out on the state of the art of ACE and ACA tools. With this state of the art in hand, an ACE challenge was proposed and executed, to analyze the subjectivity of the results of the systems, looking for the qualitative gaps in which the ACA system could act, thus improving the results.

2. Review of the State of the Art

As an initial step, we decided to conduct a state-of-the-art review of the ACE and ACA themes. Fortunately, the MIR area has a few articles focused on compiling the state of the art for these subjects.

In Mcvicar's article [1], the authors summarize the technical concepts of music that are used in ACE systems, such as chords, notes, beat times, etc. In addition, they discuss the building of ACE systems, defining a common pipeline between the observed systems. Finally, they summarize 14 years of contribution to the area, demonstrating the results of the Music Information Retrieval Evaluation eXchange (MIREX) event, which takes place annually, and results of works important to the theme of ACE. This article is being considered as the basis for the ACE research, considering the summary of the last years in the area.

In conjunction with the work above, Pauwels et al also made a study of the state of the art for chord recognition systems [2]. In their article, the work from Fujishima [3] is cited as essential to the theme, due to the proposition of a Chromagram as a 12-dimensional array, which is the standard used today. The authors demonstrate 7 existing problems in ACE activities, showing the works carried out in the last 20 years that imply these problems. Finally, they demonstrate the change in methodology that has been taking place in the MIR area, which is unifying the themes to achieve better results, citing as a recurrent problem for this change the lack of data to carry out these activities.

For the ACA theme, we can cite Dannerberg [4] as the base work. In their work, the authors define the theme of Score Alignment, and its use by computers for

automatic score alignment. In their work, they define the term automatic music accompaniment, which we call tag-along in this work. Finally, they cite some tools that did this follow-up activity and cite the potential of its use in the music industry. Hanna et al [5] carried out experiments with the theme of Chord Alignment, in which they created a system that performed the alignment of scores considering their representation by chords.

3. Running the Challenge

To collect more data for the analysis of this work, a partnership was made with the AI startup Moises.AI, to carry out a challenge. The proposal was to carry out an event in which participants could build tools for the realization of Automatic Chord Estimation using Artificial Intelligence methods, audio, and chords sheets with music as system inputs.

The expectation was that at the end of the challenge, we would have enough data and projects to quantitatively analyze the results, that is, how well they hit the chord at the correct time. It would also be possible to carry out a qualitative analysis of the results, how much they can be used by musicians or other tools for their purpose.

In addition to the challenge proposal, an evolution of the Decibel tool [6] was also released. This software allows you to carry out ACE activities through multiple sources (Audio, Chords sheets, MIDI), using available meta-information to increase the final accuracy in terms of chord and timing change. However, this tool only uses the beat time to check if the change occurred, causing the loss of information between Beats. Another point of improvement in the tool is the chord recognition functionality, which allows only Major and Minor chords. The released version, entitled DECIBEL PLUS, proposed changes in its structure to allow the use of the trained network for songs that were not in its training dataset. Another allowed change was the optional use of MIDI, as the main objective of the proposed challenge is the use of ACE with chords sheets.

As final validation, all solutions have run using a dataset containing 30 songs with varying difficulties, available in table 2 at the end of the document. Dataset information was not made available to the participants until the end of the challenge, to avoid its use in training the solutions. The metrics used were the same as MIREX [7], which uses the mir.eval library, available in Python. The main metrics chosen were CSR (accuracy) and Segmentation, using the Sevenths method available in the library.

As output, systems should return a file with the extension "LAB" in the format required by the mir.eval library mentioned above or in a JSON file containing the minimum information contained in the "LAB". The minimum information required was:

- The starting time the chord was detected;
- The end time at which the chord was detected, which must be equal to the start time of the next chord unless it is the last one in the song;

- The detected chord.

In figure 1 an example of the desired output is available. For cases where the system output occurred in JSON format, a parser would be executed to generate a file with this pattern.

Initial Time ▾	End Time ▾	Chord ▾
0.000000	2.612267	N
2.612267	11.459070	E
11.459070	12.921927	A
12.921927	17.443474	E
17.443474	20.410362	B
20.410362	21.908049	E
21.908049	23.370907	E:7/3
23.370907	24.856984	A
24.856984	26.343061	A:min/b3

Figure 1: Example of output that the system should return, opened as CSV sheet

In addition to the projects being evaluated quantitatively through the metrics mentioned above, the results were evaluated by musicians, to assess how reproducible these results were.

The challenge received entries from 11 participating teams. Of this amount, 6 teams delivered projects, but only 3 returned results within the required standards and were evaluated. The results follow in the next section.

4. Results of the Challenge

After the delivery of the projects, several environments were created to run the systems. To maintain equality between them, all systems processed the same songs and should return output in the MIREX format [7] or in a JSON format that contains all the MIREX information.

As mentioned in the previous section, the mir.eval library, used in MIREX, was used here. For its use, the LAB of Ground Truth and the LAB returned by the system as output was needed for each song and participant. Below is the result of each of the projects evaluated in all stages.

4.1. Project 1: BARÕES

For project 1, entitled BARÕES, the participants used the software released for the challenge as a basis, with changes in its execution and data entry.

System navigation occurs using a WEB interface, in which the user provided the name of the song, not the audio. Then the system searches YouTube for songs similar to the term entered and then the user should choose the one that fits the search term. Finally, the system executed its algorithm on the YouTube music and returned to the user an interface that allowed the execution of the music together with the output, allowing tag-along activity and the correction of missing or wrong chords, to then provide the final output in "LAB" format.

We verified that the Score metric was not effective, containing an average accuracy of 33% and an average segmentation of 62%, that is, 38% of the music was segmented. When analyzing the results with the tool, we verified that the accuracy was impacted due to differences in the audio used by the tool and in Ground Truth, which could be from a difference in silence at the beginning or end of the audio or different versions of the original music.

For the qualitative evaluation, the user's functionality to correct the system output in real-time, listening to the music in parallel, was well evaluated by the musicians, as this facilitates the reproduction of the result with its proper adjustments. So despite this returning a relatively high average targeting, it allowed real-time adjustments that would greatly reduce the segmentation value.

4.2. Project 2: RAGE

Project 2, entitled RAGE, used the software released for the challenge as a reference, with the addition of Sevenths chord recognition.

The system is executed in a semi-automated way. The user provides the audio as input and the system performs a search on chord sheets sites to collect the music's chords sheets, using the file name as the search point. With the audio and chord sheet available, it executes the base algorithm to return the "LAB".

This project returned an average accuracy close to 31% but had an Average Segmentation of 37%, that is, 63% of the results were segmented. When analyzing the project as a whole, we look at two possible causes for a drop in performance. The first was the addition of Sevenths chords without adding new songs in the training base that contained chords within the search spectrum, that is, new chords were added in the dictionary, but the training base continued to contain songs with only Major and Minor chords. Another problem that occurred was on the automation of the chord sheets capture because if the search returned any error, the algorithm simply did not process the song, returning 0% accuracy and 0% segmentation as result. In the scenario where the error runs were removed, the result would change to 63% average accuracy and 75% average segmentation.

For the qualitative analysis with the musicians, they realized that the results had a shift at the beginning of the song and that the tool did not capture certain chords, remaining a long time inside a chord that had already ended.

4.3. Project 3: Climber

Project 3, entitled CLIMBER used the Chordino [8] as a basis. Using the NLSS algorithm [9] from your library for chord extraction.

Its execution took place through an interface, in which the user informs the location where the songs are, chooses a song from the folder and he starts executing the algorithm. First, the algorithm runs Chordino on the audio, extracting the chords from it. Soon after, the software searches on chord sheets sites and uses these chords sheets

to reinforce the chord sequence returned by the algorithm. Finally, the system returns the result in the required LAB format.

This system returned an average accuracy of 59% and an average segmentation of 78%, but it was the only one to recognize complex chords and accidents in music, performing very well on more complex songs and failing on simpler ones. An example that we can mention is the song "Three Little Birds", in which it hit 94% of the chords, having 98% of segmentation, that is, only 2% of the segmented song.

For the analysis of the musicians, they realized that the result of the system was satisfactory, especially for more complex songs. For these, they were able to follow the music with the result of the system. An example cited was the song "A Loba", which recognized complex chords and accidents during the beats, going beyond the Sevenths chords.

4.4. Compiled results of the challenge

After all the data was collected, the results of the challenge were revealed to the participants and the dataset songs titles were made also available for verification purposes. The final results of the challenge were available in table 1 below.

4.5. Final analysis and difficulties encountered

Analyzing the results, we noticed that the systems returned similar metrics, but they all had high segmentation levels. When we took the results of the systems and put them in one software that played this music with the chords together, we noticed three behaviors.

The first was that the result usually contained a slight temporal misalignment from when the chord was played and identified by the system. This in itself would impact the performance of the song in tag-along tasks without a prior review by the user, which only Project 1 provided.

The second point that impacted the quality of the results, linked to the first, is the misidentification of chords or simply their non-recognition. In some cases, the main chords were not recognized, but their components. This was due to the chord recognition form of each software, which uses a pattern of verification by sampling. An example observed is that systems based on DECIBEL [6] recognized chords only in beat time. In more complex songs, these systems simply did not recognize chord changes between beats times, and in cases where the change happened a little before the beat time, the chord recognized was wrong due to the chromagram recognized in the exact moment containing a chord in decay.

The third point was observed only in the readings of the chords sheets in a later analysis of the results, which is the difference in music notation used between some countries. The MIREX evaluation method used in the challenge takes into account the notation of HARTE [10], but the notation used by several musicians in Brazil,

Table 1: Final Result of the Challenge

Project	CSR	Segmentation	Musician Evaluation
1	0.3291	0.6183	Good
2	0.3161	0.3739	Not Good
3	0.5877	0.7776	Good

included in most chord sheets sites in the country, is the notation proposed by Chediak [11]. This resulted in an extra effort by the musicians to understand the difference in notation. Of the 3 projects, only the last allowed the return in both patterns.

In all cases, only projects 1 and 3 were considered suitable candidates for evolution, due to their correction functionality, and recognition of complex chords, respectively. But for its use to be viable to an ACA system, it would be necessary to evolve its algorithms.

5. Conclusion and Next Steps

As noted in the results of the experiments, only three systems managed to reach the final stage. Of these, two brought results that could be used for external tasks, such as use in music accompaniment software. Even though these two systems return reasonable results, it would be necessary to evolve them so that they can perform the chord alignment activity more concretely.

As the next steps, we expect to evolve the chosen systems to experiment with techniques that focus on chord alignment. We also expect that the combination of these techniques with those of ACE will be able to bring qualitative results that can be used by music reproduction systems or by musicians and students in the tag-along activity.

For this, it is also planned to evolve the system made available in the challenge or create new independent software for chord alignment.

References

- [1] Matt Mcvicar, Yizhao Ni, and Tijl De Bie. Automatic Chord Estimation from Audio : A Review of the State of the Art. *IEEE Transactions on Audio, Speech and Language Processing*, 22(2):1–20, 2014.
- [2] Johan Pauwels, Ken O’Hanlon, Emilia Gómez, and Mark B. Sandler. 20 Years of Automatic Chord Recognition From Audio. *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019*, pages 54–63, 2019.
- [3] T Fujishima. Realtime Chord Recognition of Musical Sound: A System Using Common Lisp Music, 1999.
- [4] Roger B. Dannenberg and Christopher Raphael. Music score alignment and computer accompaniment. *Communications of the ACM*, 49(8):38–43, 2006.
- [5] Pierre Hanna, Matthias Robine, and Thomas Rocher. An alignment based system for chord sequence retrieval. *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries*, (January 2009):101–104, 2009.
- [6] Daphne Odekerken, Hendrik Vincent Koops, and Anja Volk. *DECIBEL: Improving Audio Chord Estimation for Popular Music by Alignment and Integration of Crowd-Sourced Symbolic Representations*. 2020.
- [7] Music Information Retrieval Evaluation eXchange: MIREX. https://www.music-ir.org/mirex/wiki/MIREX_HOME. Accessed: 2021-07-15.
- [8] Chordino and NNLS Chroma. <http://www.isophonics.net/nnls-chroma>. Accessed: 2021-07-15.
- [9] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, (1):135–140, 2010.
- [10] Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. *ISMIR 2005 - 6th International Conference on Music Information Retrieval*, pages 66–71, 2005.
- [11] Almir Chediak. *Dicionário de Acorde Cifrados*. Irmãos Vitale, 1984.

Table 2: Dataset Songs

ID	Artist	Song
1	Alcione	A Loba
2	Bob Dylan	Knockin on heavens door
3	Bruno e Marrone	Dormi na praça
4	Cassia Eller	Gatas Extraordinárias
5	Kid Abelha	Garotos
6	Luiz Gonzaga	Asa Branca
7	Lynyrd Skynyrd	Sweet home alabama
8	Marília Mendonça	Alô porteiro
9	The Black Eyed Peas	I gotta feeling
10	Nando Reis	Por onde andei
11	Aline Barros	Digno é o Senhor
12	Barão Vermelho	Bete balanço
13	Bob Marley	Three Little Birds
14	Capital Inicial	Primeiros erros
15	Cassia Eller	Malandragem
16	Creedence Clearwater Revival	Have you ever seen the rain
17	Jorge e Mateus	Louca de saudade
18	Legião Urbana	Quase sem querer
19	Legião Urbana	Pais e filhos
20	Luiz Gonzaga	Derramaro o gai
21	Marília Mendonça	Eu sei de cor
22	Maroon 5	Animals
23	Melim	Ouvi dizer
24	Melim	Meu abrigo
25	Nando Reis	Relicario
26	Pink Floyd	Wish you were here
27	Roupa Nova	Os corações não são iguais
28	The Beatles	Eleanor Rigby
29	Titãs	Porque eu sei que é amor
30	Victor e Leo	Borboletas

Training a convolutional neural network for note onset detection on the clarinet

Tairone N. Magalhães¹, Mauricio A. Loureiro¹

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

tairone@ufmg.br, mauricioloureiro@ufmg.br

Abstract. *Although computational models for note onset detection have improved drastically in the last decade, mainly due to the advances brought by the field of Deep Learning, such models have not been perfected yet. When dealing with specific data, like clarinet recordings, those models still produce a significant number of false positives and negatives. In this paper, we evaluate pre-trained onset detection models from the library *madmom* on a dataset composed of solo clarinet recordings, in particular, to investigate their performance on this kind of data. Moreover, we use the clarinet dataset to train the same neural network (CNN) employed in one of those models, to investigate whether training the model on this specific data leads to an improvement when dealing with clarinet recordings. The results obtained from the model trained strictly on clarinet data are considerably better than those from models trained on generic data.*

1. Introduction

Note onset detection consists in finding, in audio signals, the instants at which musical notes start. The automatic detection of note onsets is a difficult task. Although some of the onset detection methods/models currently available produce excellent results in recordings of instruments with prominent note attack, their performance can drop significantly when dealing with notes with very soft attack, like woodwind instruments, for example. Clarinet recordings seem to be particularly difficult, compared to other instruments.

Many studies on automatic onset detection have been published since the beginning of the 2000s. In the beginning, the onset detection methods were mostly based on DSP techniques [1, 2, 3, 4, 5]. Recently, the majority of the studies have been employing machine learning, producing models that achieve better results on the task. However, most of these models are trained on mixed datasets, frequently containing recordings of several different instruments and musical genres. Since machine learning algorithms learn to perform tasks based on patterns that are extracted from input data, the quality of the training dataset has a pivotal impact on the quality of the final model. Having a good dataset that captures the characteristics of “real world” data is key to obtaining a model with good performance. Although it is important to avoid biases in the dataset that may negatively impact the model’s generalization capacity, in some circumstances, a model might benefit from specialized data that is biased towards the expected final purpose of the model.

This led us to question whether training a machine learning model on a dataset that contains only clarinet recordings would produce a model that performed better on clarinet onsets. In this paper we test and evaluate three onset detection models from the library *madmom* [6] on clarinet recordings, specifically. Two of those models are based on neural networks, a convolutional neural network (CNN) and a recurrent neural network (RNN), and were previously trained by the author of the library on a generic dataset. The other method uses strictly DSP techniques and is based on spectral flux. We also test and evaluate a model trained by ourselves using a clarinet dataset, specifically. This model uses the same CNN architecture from *madmom*’s model. Thus, we can check whether this specific training data leads to a better note onset detection model for the instrument.

2. Methodology

To obtain robust segmentation models using machine learning, it is necessary to provide a dataset with labeled target values. A labeled dataset also allows us to evaluate the models and compare the results they produce. Therefore, in this section we first introduce the two datasets used in our experiments. Then we detail the onset detection models employed in our experiments, followed by an explanation of the evaluation criteria we adopted.

2.1. Datasets

We created two different datasets for this experiment, both containing strictly solo clarinet recordings. To annotate the note onsets, we used a specialized web application for audio annotation developed by ourselves, called *Audio Segment Annotator*¹. This tool seeks to provide a fine temporal resolution of 2 milliseconds, in a user interface with a steep learning curve and straightforward controls. It provides visualizations of the audio waveform and its spectrogram, synchronized with respect to the time axis.

2.1.1. Clari-onsets-50

Most datasets available made available by researchers for training note onset detection models do not focus on monophonic recordings or do not contain a significant amount of clarinet onsets. Therefore, we decided to create a large dataset composed exclusively of clarinet recordings. This enables us to train, validate and test a specialized onset detection model for clarinet. We refer to this dataset as

¹This tool was developed for internal use in our laboratory, and has not been published yet.

absolute error (ms)	count	percentage	cumulative percentage
0 to 4 ms	541	85.9	85.9
4 to 8 ms	67	10.6	96.5
8 to 12 ms	16	2.5	99.0
12 to 16 ms	4	0.6	99.7
16 to 20 ms	1	0.2	99.8
> 20 ms	1	0.2	100.0

Table 1: Absolute error for the manual annotations obtained for the dataset *clari-onsets-3*.

clari-onsets-50. It is composed by 50 excerpts of solo clarinet recordings, totaling 23 minutes and 6 seconds of audio, with a mean duration of 27.7 seconds. Half of those recordings were made in our laboratories, while the other half was obtained from commercial recordings from several different albums and clarinetists. They comprised a few different musical genres, including classical, jazz, and contemporary pieces. The recordings are very diverse with respect to level of reverberation, distance of the microphone, background noise, etc. This was intentional, since a diverse dataset, with respect to those characteristics, tend to lead to a model that is able to generalize better when the characteristics of the audio change. The excerpts were all annotated by a single person, without repetition, over several weeks. A total of 3551 note onsets were obtained for the 50 excerpts.

2.1.2. Clari-onsets-3

The dataset *clari-onsets-3* contains only three recordings, with a total of 126 note onsets. The recordings were all made in the laboratory, in a room with very little reverberation. The excerpts were chosen seeking to achieve a diversity of musical material, with notes in different registers of the instrument, melodies containing both short and long intervals, notes with soft and sharp attacks, and different dynamics. For this particular dataset, we repeated the onset annotations five times (all by the same person), seeking to obtain: (1) a reliable estimation of the note onset time, based on multiple measurements; and (2) an estimation of the measurement error for the task. The annotation errors were estimated by calculating the time difference between each annotated onset (from the different sessions) and the median, and are shown in table 1. This dataset is rather small, and was not used for training any neural network, but we used it as a test dataset in our experiments.

2.2. Onset Detection Models

2.2.1. CNN

This method was proposed by Schlüter and Böck, and is based on a convolutional neural network applied to mel-scaled spectrograms [7]. The advantage of this representation is that it uses a logarithmic scale for the frequency bins instead of a linear one. This logarithmic scale models better the human perception of pitch, allowing the spectrogram to have a better perceptual pitch resolution using a relatively

small number of bins, thus reducing the model’s memory demand.

The input of the model consists of three stacked 80-band mel spectrograms with logarithmically scaled magnitudes, each calculated using a different window length: 1024 (23 ms), 2048 (46 ms), and 4096 (93 ms) samples. They are calculated using a hop length of 441 samples, which corresponds to a frame resolution of 10 ms. The input of the network consists of a group of 15 contiguous spectral frames centered on the frame to be classified. So each example processed by the network corresponds to a context of 150 ms. The onset annotations (which consist of time points) are converted to target values, which are defined for each input example, using the criteria described below.

- If there is an onset annotation within the 10 ms time window corresponding to the central frame for a given input, the target value is assigned to 1 (onset).
- Since the annotation error might be a bit larger than this window, each frame immediately before or after a frame that was set to 1 in the step above is also set to 1, giving a margin for the network to learn from annotations that are not precise (fuzziness). These samples are also weighted with a factor of 0.25 during training.
- The targets for any other frames are defined to 0 (no onset).

Basically, the architecture of the network consists of two convolutional layers (2D convolutions) followed by an intermediary fully-connected layer, and the output layer, which is also fully-connected. The network architecture is composed by the following layers:

- input: 15 frames x 80 bands;
- 2D convolutional layer: filters of 7 frames x 3 bands, computing 10 feature maps;
- max-pooling over 3 bands with no overlap;
- 2D convolutional layer: filters of 3 x 3, computing 20 feature maps;
- max-pooling over 3 bands with no overlap;
- fully connected layer with 256 units;
- fully connected layer with a single unit (output).

All the layers use rectified linear unit activation functions. Dropout is applied to the inputs of the fully connected layers for regularization. The network is optimized using gradient descent, minimizing binary cross-entropy loss. It is trained for 100 epochs, using a starting learning rate of 1.0, which is multiplied by 0.995 at the end of each epoch. The initial momentum is 0.45, and it is linearly increased to 0.9, between epochs 10 and 20. The mini-batches used for each training step consist in 256 examples.

The network produces output values between 0 and 1, representing the frame-wise probability of onset along the audio signal. To obtain the onset points, first, this time series is convoluted with a Hamming window of 5 frames, which smooths it a bit. Then a peak-picking

method is used to extract the local peaks higher than a given threshold value. This threshold is chosen by varying its value and picking the one that produces the highest F-score. The instants of the peaks correspond to the onsets detected by the model.

2.2.2. RNN

The model proposed by Eyben et al. [8] uses as input a stacked array of features consisting in: (1) a mel spectrogram calculated using a window length of 1024 samples (23 ms); a mel spectrogram computed using a window length of 2048 samples (46 ms); (3) the positive first-order difference between two successive mel spectrogram frames at the 1024 samples window length; (4) the positive first-order difference at the 2048 samples window length. The positive first-order difference is calculated by applying a half-wave rectifier function $H(x) = \frac{x+|x|}{2}$ to the difference between two consecutive mel spectrogram bands. A total of 40 mel bands are used in this model, achieving 160 stacked input features for each time step. The input features are calculated using a sliding window with a hop length of 441 samples (10 ms). Each hop performed by the sliding window corresponds to one time step in the model.

The model consists of a bidirectional long short-term memory network (LSTM) applied to the described input features. Bidirectional recurrent networks incorporate future context into a network by adding an extra layer for each hidden layer, which will process the input sequence backward, while the other layer processes it forwards. It produces a non-causal model that outputs values that depend on both the past and future time steps. The network contains 6 hidden layers, 3 of them for processing the input forwardly and 3 backwardly. Each hidden layer contains 20 LSTM units, and the output layer has two units, which are normalized to sum up to 1.0, using the softmax function. The outputs represent the probabilities of onset and no onset. The authors reported that they used this two-output approach because the results using a single output were not as successful. To obtain the final onset times, a peak picking method is used to detect the local maxima just for the output corresponding to the onset class.

2.2.3. SuperFlux

One of the most popular DSP-based methods for note onset detection is based on calculating the spectral flux, based on the idea that note onsets are accompanied by rapid spectral changes in the audio signal. The *SuperFlux* method, proposed by Böck and Widmer [9], consists in adding some improvements to the spectral flux method. Instead of calculating the bin-wise difference between consecutive spectral frames, it adds a trajectory-tracking stage to the method to avoid high values in the ODF. It also combines phase information, using a technique called local group delay, to reduce the impact of amplitude variations that occur in steady tones in the output of the method. These changes seek to reduce false positives caused by *vibrato* and *tremolo*.

2.2.4. Clarinet-specific model

To obtain a specialized note onset detector for clarinet recordings, we implemented the CNN architecture de-

scribed in section 2.2.1 and trained this network on a clarinet dataset. In the original paper, by Schlüter and Böck [7], their model had been trained on a dataset containing both monophonic and polyphonic recordings played on various instruments and covering multiple musical genres. We anticipated that that model would not perform as well as it could on clarinet recordings due to the specific characteristics of the sound produced by the instrument, which are underrepresented in the training data.

We adjusted a few hyperparameters in our model. The learning rate was set to 0.1, and the weights of the frames that precede and succeed the annotated onset were set to 0.4 (fuzziness). These hyperparameters were adjusted by testing different values and evaluating the result on a validation set.

2.3. Evaluation Criteria

To evaluate a note onset detection model, we need to define criteria to decide whether each onset prediction is correct or not. Thus, the predictions must be compared and matched to the onset annotations (ground truth). For this purpose, we specified an argument ω , which is a tolerance window (defined in milliseconds) centered on each annotated value. To be considered valid, a prediction must fall within the defined tolerance window around the onset annotation. When two onset predictions fall within the tolerance window around a single annotation, only the closest one is accounted as a true positive, while the other is considered a false positive. With those values, we can calculate the metrics precision, recall, and F-score, to evaluate the results of an onset detection method. All the results presented in this paper show the evaluation metrics calculated using two different values for the tolerance window ω : 20 ms and 50 ms. Since these tolerance windows are centered on the target onset point, an onset detection that is within their limits will correspond to a maximum distance of 10 ms and 25 ms, respectively.

3. Results

The following names are used to refer to the models we evaluated in our experiments:

- CNN-Böck
- RNN-Böck
- SuperFlux
- CNN-Clari

The models *CNN-Böck* and *RNN-Böck* were pre-trained on a generic dataset, known as Böck dataset, containing 102 minutes of audio and 25,927 onsets composed by multiple instruments and musical genres, with both polyphonic and monophonic recordings. The model *Superflux* is based on DSP techniques. *CNN-Clari* was trained by ourselves using the dataset *clari-onsets-50*.

Table 2 shows the results obtained using each of the models on the dataset *clari-onsets-50*. To train and evaluate the model *CNN-Clari* using the same dataset, we employed 10-fold cross-validation, using 8 folds for training, 1 for validation, and 1 for testing. The model *CNN-Clari* generated the best results among the four models,

method	$\omega = 50$ ms			$\omega = 20$ ms		
	f-score	recall	precision	f-score	recall	precision
CNN-Böck	0.861	0.851	0.872	0.605	0.597	0.614
RNN-Böck	0.781	0.779	0.784	0.459	0.456	0.461
SuperFlux	0.718	0.669	0.775	0.379	0.353	0.410
CNN-Clari	0.954	0.946	0.962	0.720	0.712	0.728

Table 2: Results obtained for the dataset *clari-onsets-50*. The model CNN-Clari was trained and evaluated using 10-fold cross-validation, using 8 folds for training, 1 for validation, and 1 for testing.

method	$\omega = 50$ ms			$\omega = 20$ ms		
	f-score	recall	precision	f-score	recall	precision
CNN-Böck	0.948	0.944	0.952	0.709	0.706	0.712
RNN-Böck	0.887	0.905	0.870	0.685	0.698	0.672
SuperFlux	0.862	0.889	0.836	0.469	0.484	0.455
CNN-Clari	0.988	0.984	0.992	0.908	0.905	0.912

Table 3: Results obtained for the dataset *clari-onsets-3*. This time, the model CNN-Clari was trained on the entire dataset *clari-onsets-50*, and the resulting model was employed on the recordings from *clari-onsets-3*.

for tolerance windows of both 50 ms and 20 ms, with respect to all the three metrics: F-score, recall, and precision. The F-score obtained for the 50 ms tolerance window was 0.954, more than nine percentual points higher than the second best F-score. Among the models from the library madmom, *CNN-Böck* was the best, reaching an F-score of 0.861 against 0.781 from *RNN-Böck* and 0.718 from *SuperFlux*. This result corroborates with the results obtained in [7] when comparing these three models, in which the CNN had performed better than the other two models, on a generic dataset. Nonetheless, the results

The results obtained for the dataset *clari-onsets-3* are shown in table 3. For this particular experiment, we trained the model *CNN-Clari* using the entire dataset *clari-onsets-50*.

Again, the model *CNN-Clari* produced the best results, with an F-score of 0.988 for the 50 ms tolerance window, followed by *CNN-Böck* (0.948), *RNN-Böck* (0.887) and *SuperFlux* (0.862). The results obtained for this smaller dataset were considerably better than for *clari-onsets-50*.

4. Discussion

Training a CNN model on a clarinet dataset, specifically, we were able to achieve results that are significantly better, for this particular domain, than those produced by models trained on generic data or by a model that uses strictly DSP techniques. If we order the results from the models based on the resulting F-score, we obtain the sequence *CNN-Clari*, *CNN-Böck*, *RNN-Böck* and *SuperFlux*. This sequence is consistent among the results from both experiments and both tolerance windows.

Regarding the dataset *clari-onsets-3*, for which we obtained results that are considerably better when compared to *clari-onsets-50*, it is worth emphasizing that it is a small dataset, having a total duration that corresponds to only 3.4% percent of the latter, and there is certainly a bias

related to the fact that its recordings were all made in a laboratory, under controlled conditions, in a room with very low reverberation, by only two clarinetists. The low reverberation is probably a factor that makes it a lot easier for any method to detect the onsets since it minimizes the superposition of energy of consecutive notes.

By listening to the note onsets detected by the clarinet-specific model, we observed that it tended to miss some onset for notes with extremely long attack times (very soft notes), producing false negatives. During the dataset annotation we had already noticed that it was quite hard to determine the exact position of the onset for such notes. Thus, we suspect that those false negatives might have some relation to the annotated data's imprecision for soft notes. Also, on a few situations where there were two consecutive notes with the same pitch, the model tended to miss the second note's onset.

In the clarinet recordings we used, there is a significant amount of breathing sounds, background noises, crackling sounds from the chair, and sometimes even whisper sounds. It is interesting to notice that the network was able to learn to ignore those sounds completely, which is an excellent result, since they would not correspond to note onsets in the vast majority of the clarinet repertoire (although they might, in contemporary compositions).

It is worth mentioning that we did some preliminary experiments training the clarinet onset detection model using additional semi-synthetic data, but so far, the results have not been promising. That data was created using a MIDI keyboard controller to record monophonic melodies using the sound from a commercial clarinet sampler software. The *note on* events of the MIDI protocol were used to generate the onset annotations, and those were used in our training experiments. We attempted to use this semi-synthetic data in the CNN model using two different strategies: (1) employing them on a pre-training phase and (2) mixing them in the training data. None of these approaches

improved the results of the model. As a matter of fact, they produced slightly worse results than the models trained without them. Yet, these results are preliminary and still require further investigation. We also did some experiments applying data augmentation techniques to the train data (such as pitch-shifting, time-stretching, gain adjustment, and addition of noise). Again, in our preliminary experiments using these techniques, the results of the model got slightly worse. One thing that might be worth further investigation are the false negatives produced by the model *CNN-Clari*, specially on notes with soft attack. As shown in tables 2 and 3, the recall for the model *CNN-Clari* is slightly lower than its precision, so if we figure out a way to reduce those false negatives, we may get a significant improvement in the results produced by the model.

Acknowledgements

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] Chris Duxbury, Mark Sandler, and Mike Davies. A hybrid approach to musical note onset detection. *Computer*, pages 33–38, 2002.
- [2] 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.
- [3] Nick Collins. Using a pitch detector for onset detection. *Proceedings of the International Symposium on Music Information Retrieval*, pages 100–106, 2005.
- [4] Ruohua Zhou and J.D. Reiss. Music onset detection combining energy-based and pitch-based approaches. *Proc. MIREX Audio Onset Detection Contest*, 2007.
- [5] Harvey Thornburg, Randal J. Leistikow, and Jonathan Berger. Melody extraction and musical onset detection via probabilistic models of framewise STFT peak data. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1257–1272, 2007.
- [6] Sebastian Böck, Filip Korzeniowski, Jan Schlüter, Florian Krebs, and Gerhard Widmer. madmom: a new Python Audio and Music Signal Processing Library. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 1174–1178, Amsterdam, The Netherlands, 2016.
- [7] 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, Prague, Czech Republic, may 2014. IEEE.
- [8] Florian Eyben, Sebastian Böck, Björn Schuller, and Alex Graves. Universal Onset Detection with Bidirectional Long-Short Term Memory Neural Networks. *Proceedings 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, (June 2017):589–594, 2010.
- [9] Sebastian Böck and Gerhard Widmer. Maximum Filter Vibrato Suppression for Onset Detection. In *Proc. of the 16th Int. Conference on Digital Audio Effects (DAFx-13)*, Maynooth, Ireland, September 2-5, 2013, pages 1–7, 2013.

On Generalist and Domain-Specific Music Classification Models and Their Impacts on Brazilian Music Genre Recognition

Diego Furtado Silva¹, Angelo Cesar Mendes da Silva²,
Luís Felipe Ortolan¹, Ricardo Marcondes Marcacini²

¹Departamento de Computação – Universidade Federal de São Carlos, Brazil
Rod. Washington Luís, km 235, Jardim Guanabara – 13565-905 São Carlos, SP

²Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, Brazil
Av Trabalhador São-carlense, 400, Centro – 13566-590 São Carlos, SP

diegoofs@ufscar.br, luisfelipe.ortolan@gmail.com, {angelo.mendes, ricardo.marcacini}@usp.br

Abstract. Deep learning has become the standard procedure to deal with Music Information Retrieval problems. This category of machine learning algorithms has achieved state-of-the-art results in several tasks, such as classification and auto-tagging. However, obtaining a good-performing model requires a significant amount of data. At the same time, most of the music datasets available lack cultural diversity. Therefore, the performance of the currently most used pre-trained models on underrepresented music genres is unknown. If music models follow the same direction that language models in Natural Language Processing, they should have poorer performance on music styles that are not present in the data used to train them. To verify this assumption, we use a well-known music model designed for auto-tagging in the task of genre recognition. We trained this model from scratch using a large general-domain dataset and two subsets specifying different domains. We empirically show that models trained on specific-domain data perform better than generalist models to classify music in the same domain, even trained with a smaller dataset. This outcome is distinctly observed in the subset that mainly contains Brazilian music, including several usually underrepresented genres.

1. Introduction

Music classification is one of the most investigated subtasks of Music Information Retrieval (MIR) [1]. Like other domains that rely on complex data, such as Computer Vision (CV) and Natural Language Processing (NLP), content-based Music classification has achieved significant improvements through deep neural networks [2].

A limitation of deep learning techniques is that they usually need a large volume of training data to obtain a good-performing model. Therefore, applying deep learning requires a significant effort of labeling the data or, at least, crowdsourcing the labeling responsibility. Both cases lead to a more severe difficulty for training models for underrepresented music genres in music classification.

A possible solution to circumvent this problem is the use of pre-trained models [3, 4]. In this scenario, we use a model trained in a large dataset and fine-tune its weights using the dataset we have at hand. This procedure, called transfer learning [5], is increasingly becoming standard in CV, NLP, and MIR applications. Besides, it is possible to use the pre-trained model in some scenarios with

no fine-tuning. This procedure is called zero-shot learning [6]. Finally, recent advancements in self-supervised learning have led researchers to create models where other researchers or practitioners may use the model as a feature extractor and apply the learned representation in different downstream tasks.

However, several efforts on CV and NLP have shown that the transference of knowledge is limited, mainly when we use a pre-trained model in a completely different application domain [7, 8]. Usually, fine-tuning from a domain-specific model (similar to the one it will be applied) leads to better results than generalist models.

Although pre-trained models and their ability to transfer knowledge are well-studied on CV and NLP applications, these techniques are less explored in MIR applications. A few works have used CV-based models, like VGG, to classify music data. More recent papers achieved state-of-the-art music classification using musically motivated models, creating evidence that supports domain-specific models for music data [9]. However, there is no effort to assess the capabilities of domain-specific or general music classification models on low-represented music genres.

This work presents a study of the impact of domain-specific training of music models for Multi-label Music Genre Classification. Notably, we investigate how a widely-used deep architecture behaves on different Western music genres when trained using songs in other groups. For example, we assume that songs composed in English usually have significant differences from songs written in Portuguese, so we apply different language-based filters to train general and specific models.

More importantly, the portion of our dataset composed in Portuguese includes Brazilian genres usually low-represented in music datasets, such as samba, sertanejo, and axé. Moreover, we also understand that songs in the same genre, such as rock and reggae, may present different characteristics if written in English or Portuguese, and the domain-specific models should capture these differences.

2. Background and Related Work

In the last few decades, we have witnessed a significant increase in Machine Learning applications in Computer Music and Music Information Retrieval tasks [10, 11, 1]. The

most common approach in this context is using a supervised learning algorithm. This category of techniques may be applied in a plethora of tasks, such as genre classification [12, 13], auto-tagging [14, 4], instrument identification [15], mood [16] and emotion classification [17], chord recognition [18], and popularity estimation [19].

Early systems usually applied a feature extraction step to obtain a structured dataset before submitting it to “traditional” Machine Learning algorithms [20]. For instance, different features like signal or spectrum parameters, mel-frequency cepstrum coefficients (MFCC), and psychoacoustic characteristics [21] may be extracted using different windowing and aggregation strategies from the raw audio in genre classification [22, 23, 12]. However, aggregating the windows and the influence of distinct features and their combinations make music classification difficult.

For this reason, the community has moved to deep learning algorithms [3, 2]. As a result, most of the literature that defines the state-of-the-art of music classification is based on learning a deep neural network-based model [3, 24, 25, 26, 27, 4, 28, 13]. The main advantage of using these models is the ability to learn the best representation to solve the problem at hand.

Several recent works have used neural models trained in a dataset to deal with another dataset, especially with only a few labeled examples. For example, we may rely on transfer learning [5, 29, 30] or zero-shot [6] learning strategies to use these models. Moreover, we may leverage on different supervision levels, such as supervised [31] or self-supervised learning [32, 33] to learn these models.

Considering the resemblance of these strategies to how researchers and practitioners have used *language models* to approach NLP problems, we refer to these models for MIR tasks as *music models*. Among the various music models found in the literature, we highlight *musicnn* [4], which has received considerable attention.

*musicnn*¹ is a set of pre-trained musically motivated convolutional neural networks designed for music auto-tagging. A researcher interested in using this music model may train it from scratch, fine-tune its weights, or use it as a feature extractor. Its default output consists of a taggram consisting of each pre-defined tag’s strength (prediction score) in each time window of the input recording. Alternatively, the user may extract the penultimate layers of the model and use them as low-level features.

One limitation of these music models, including *musicnn*, is that they are majorly trained using American-like music, such as pop, hip-hop, and blues songs. As a result, when a song from a different culture is presented to the model, it may fail in extracting suitable features to describe it. For instance, Brazilian music is rich in influences worldwide. Consequently, Brazilian musicians have created several regional genres, such as forró,

axé, samba, funk carioca, bossa nova, and sertanejo. Unfortunately, the *musicnn* models have no tags regarding these genres and the instruments used to record most of the songs in them.

A few papers noticed the lack of Brazilian music in the MIR literature. For instance, Conceição et al. [34] studied the most used music classification datasets and stated a lack of cultural diversity in them. So, the authors created a database containing 613 Brazilian songs from 6 genres. Using low-level feature extraction and traditional Machine Learning algorithms, the authors achieved slightly better results using only the Brazilian songs instead of merging them with a public dataset.

It was not the first time authors approached the lack of cultural diversity on music datasets, especially considering Brazilian or Latin music. For instance, the Latin Music Database (LMD) [35] was proposed to fill this lack. The LMD contains features extracted from 3160 songs in 10 Latin genres, such as salsa and samba. Later, de Sousa, Pereira, and Veloso [36] proposed the Brazilian Music Dataset, containing 120 songs labeled in 7 musical genres. Other datasets of Brazilian genres were presented for different tasks, such as the Brazil Northeast data [37], the Brazilian Popular Music [38], the SambaSet [39], and the Forró em Vinil Dataset [40].

Recently, the 4MuLA dataset was proposed: A Multitask, Multimodal, and Multilingual Dataset of Music Lyrics and Audio Features [41] which has richer musical information than the datasets aforementioned. 4MuLA makes available 96,458 Brazilian and international songs that belong to 15,310 artists and are organized into 76 genres in its full version. The collected songs were from the Brazilian portal Vagalume. All pieces have a set of pre-processed metadata, lyrics, and acoustic features to represent them in different scenarios.

3. Domain-Specific Models

This work is based on the widely spread idea that domain-specific deep models usually perform better than generalist ones. This idea is remarkably studied on Natural Language Processing (NLP) tasks. The current state-of-the-art NLP algorithms are based on pre-trained language models, such as Bidirectional Encoder Representations from Transformers (BERT) [42].

However, BERT (and other language models) present several flaws identified in the literature. For instance, multilingual models have deficiencies in modeling underrepresented languages [8]. Usually, replacing the multilingual model with a specialized monolingual version improves the performance on different NLP tasks [43].

Besides, training the language model on texts of a specific domain also leads to better results. For this reason, the literature presents a vast diversity of domain-specific BERT. Some examples of knowledge domains used to train domain-specific language models are biomedicine [44], finance [45], and scientific writing [46].

¹<https://github.com/jordipons/musicnn>

This work hypothesizes that music models behave like language models for domain-specific data. To evaluate this hypothesis, we use a dataset containing several songs of Brazilian music. We also use a similar dataset containing other songs concentrated on American music, but not limited to. For this, we separate songs from the 4MuLA dataset [41] composed in Portuguese and English. The complete dataset, which contains these and other languages, is used to train a generalist model.

These datasets are used to train `musicnn` [4], a musically motivated convolutional neural network designed for auto-tagging. Using the different datasets, we will evaluate how `musicnn` behaves on various combinations of general and specific-domain training and test.

Figure 1 describes the `musicnn`'s architecture. The input of the deep network is a log-scaled mel-spectrogram. First, it is submitted to musically motivated convolutional layers [47], creating the first representation that may be used, named by the authors as front-end features. Next, these features are passed through layers of fully connected (dense) unities, transforming the front-end into mid-end features. Finally, after pooling and dense layers, the network output a taggram.

The taggram represents the predicted score of each of the 50 tags comprised in the training dataset in short time windows. In this work, we are interested in the genres of the assessed songs. So, we substitute the last layer to adapt it to the number of genres comprised by each subset (general or domain-specific). Then, we used the mean value, across all the windows, for each genre as the final scores.

4. Experimental Setup

We used the 4MuLA dataset in our experiments due to the completeness of its information. We chose to use the small version, which concentrates songs from artists in the top 100 ranking positions according to the source of data Vagalume website². In this version, each audio has 30 seconds and is represented by a melspectrogram. The 4MuLA small version has 9661 songs that belong to 419 artists and is organized in 50 different genres.

To obtain the domain-specific models, we first needed to define the rule to specify each domain. For this, we explore the main characteristics of the dataset. Table 1 show a summary of small 4MuLA. We note that each song in the dataset contains a top genre and (possibly) a list of other/secondary genres.

While we could use specific genres, we opted to use the language the lyrics were composed as a filter. We understand that it brings sufficient information to define the domains to verify the quality of the studied model on Brazilian music. Besides, considering English and Portuguese as the filters, we create two specific-domain datasets with a similar number of instances.

²<https://www.vagalume.com.br/>

Table 1: Overview of 4MuLA small

Feature	Amount
Number of songs/instances	9661
Number of songs (EN)	4722
Number of songs (PT)	4654
Number of songs (others)	285
Unique artists	491
Unique genres	50
Max instances in a single "top" genre	1169
Min instances in a single "top" genre	11
Avg instances in a single "top" genre	189.43
Std instances in a single "top" genre	260.86

On the other hand, we recognize that we may find Bossa Nova and other Brazilian genres written in English. However, filtering by language also has the impact of separating songs of the same genre with different influences. For instance, Brazilian rock music may reflect a significant influence from local music. So, American or British rock music, for example, may sound considerably different from Brazilian songs in the same genre.

Therefore, we created three sub-datasets. The first one uses the whole 4MuLA small, comprising songs from all the languages and, consequently, varied cultures. We refer to this dataset and the models created from it using the tag `All`. From this dataset, we created random training and test partitions, named as `All-train` and `All-test`, respectively.

In a second stage, we used `All-train` and `All-test` to create the domain-specific data. By filtering the data regarding songs written in Portuguese, we obtained the datasets `PT-train` and `PT-test`. Similarly, we applied the filter for English and obtained the datasets `EN-train` and `EN-test`. Figure 2 illustrates this procedure.

Finally, we recall we use `musicnn` as the base model of our evaluation. While the currently available `musicnn` may be used as a pre-trained model, we opted for initially training it from scratch. This option relies on avoiding possible intersections of the songs in our dataset and the datasets used to train `musicnn`, as well as the possibility of the pre-trained model already reflects some characteristics of our domain-specific data. Moreover, we consider the number of examples in 4MuLA small as sufficient for training a suitable model.

Therefore, we used `All-train` to create our pre-trained model, which we refer to as `musicnn-All`. For the sake of comparison, we also trained specific-domain models from scratch. Using the datasets `PT-train` and `EN-train`, we trained the models `musicnn-PT` and `musicnn-EN`, respectively.

Finally, we noticed that we repeated the experiments ten times to present results robust to variations caused by randomness.

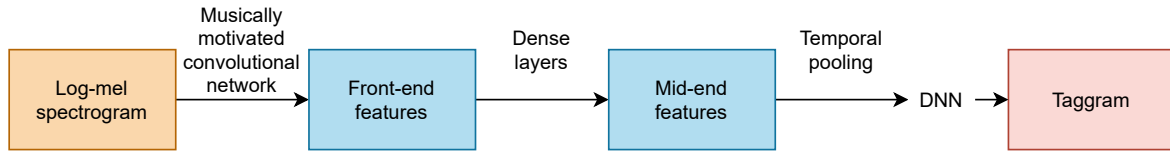


Figure 1: General architecture of musicnn, the model we used as based in our experiments

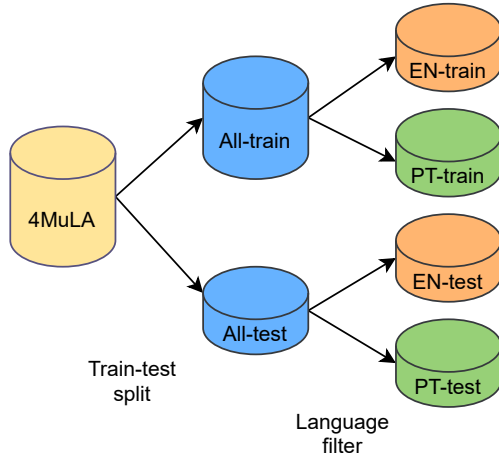


Figure 2: We split our dataset in train and test partitions and apply filter on these subsets to obtain domain-specific data

5. Results and Discussion

Using the described experimental setup, we assessed the area under the Receiver Operating Characteristic Curve (ROC-AUC) and the Precision-Recall (PR-AUC) curves obtained by each model on each test set. Table 2 presents the average results.

Table 2: Detailed results

Model	Test set	ROC-AUC	PR-AUC
musicnn-All	All-test	0.68	0.43
	EN-test	0.65	0.45
	PT-test	0.64	0.34
musicnn-EN	All-test	0.60	0.36
	EN-test	0.65	0.44
	PT-test	0.53	0.27
musicnn-PT	All-test	0.63	0.39
	EN-test	0.56	0.37
	PT-test	0.69	0.39

We plot these results differently aggregated in separate charts for visualization and, consequently, better interpretation of the results. Figure 3a presents the ROC-AUC performances aggregated by model so that we can compare their performances for each language filter. Figure 3b aggregates the results by language filter.

When observing the ROC-AUC, we note that the best performance of each model occurs on the test set obtained by the same filter used in the training set. This be-

havior is expected for musicnn-EN and musicnn-PT for several reasons. We can interpret this result as anticipated evidence that zero-shot learning does not work when the pre-trained models are domain-specific, especially if applied on another domain-specific dataset. For instance, PT-test comprises several songs from genres not included in EN-train. Even the genres included in both datasets may present differences on the separate data, as previously discussed. So, these songs appear on both generalist and specific (for the other language) test sets. Once these songs represent a fraction of All-test, the performance decrease is lower on this data.

On the other hand, the model trained using all data brings more interesting results. While we could expect better results on PT-test and EN-test, the best performance of musicnn-All was achieved on the All-test dataset. It means that, for some reason, musicnn-All performed well on songs that are neither in PT-train nor in EN-train.

Regarding each test dataset, using a model trained on a domain generally leads to better ROC-AUC rates for the same domain. This fact is clearly observed for All-test and PT-test. For instance, musicnn-PT is five percentage points better than musicnn-All for PT-test. However, the results obtained by the generalist and specific models are similar on EN-test.

Besides, musicnn-PT performs better than musicnn-EN on All-test. Since the subsets are not unbalanced, we can conclude that the songs comprised by PT-train lead to a better model (for general music) than the songs on EN-train. It possibly happens because the Portuguese songs, at least for this dataset, are more diverse.

When we observe the PR-AUC results, we obtain similar conclusions. However, there is a slight inversion in the results obtained on EN-test. The best PR rate on EN-test was obtained by musicnn-All (Figure 4b). At the same time, the highest PR-AUC obtained by musicnn-All was on EN-test (Figure 4a), not on All-test like for the ROC-AUC.

A general observation made on the results is that all the ROC-AUC are significantly higher than the PR-AUC. Won et al. [9] shown the same phenomenon when comparing diverse deep learning models for music auto-tagging. Moreover, they observed some inversions of result interpretations, like those we discussed for EN-test. However, the authors did not propose any solution for a better evaluation.

We added all the confusion matrices obtained in

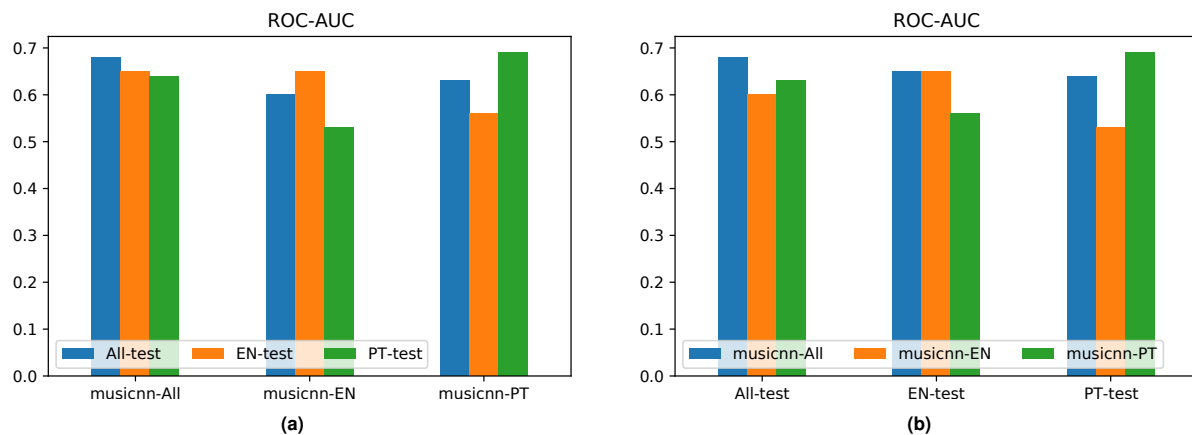


Figure 3: ROC-AUC obtained in our experimental evaluation, aggregated by model (a) and dataset (b)

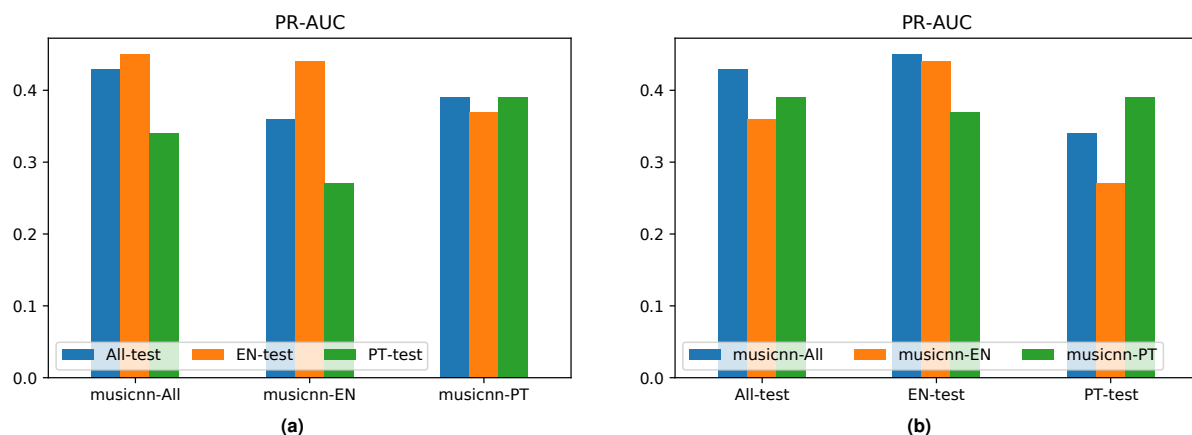


Figure 4: PR-AUC obtained in our experimental evaluation, aggregated by model (a) and dataset (b)

our experimental evaluation in the appendix to a better and detailed interpretation of the results. Carefully analyzing these matrices helps us interpret some outcomes that are not evident in the bar plots and tables. For instance, we can observe that our models obtain a significant number of false negatives and only a few false positives, causing low recall rates. It explains why the PR-AUC values are notably lower than the ROC-AUC.

6. Final Remarks

We presented an experiment on how a known music model behaves in general and specific domains. Notably, we filtered songs by the language of their lyrics as a heuristic to create a dataset containing several recordings of Brazilian music and another one more similar to well-known datasets, where the songs were composed in English.

Our results show that specific-domain models, in general, are more suitable to classify domain-specific data. Training a model from scratch using only the Portuguese portion of the data drove us to build better models for the same type of songs than training the same model using the same data and, additionally, data from other languages. At the same time, it is not clear for songs composed in English, which leads us to think that Brazilian music's complexity played a significant role in the model training.

Although the evaluated model performs similarly

to other deep learning models for music classification [9], we intend to perform similar experiments using other architectures. Moreover, establishing and evaluating different strategies to define the specific domains is also left to future work.

Furthermore, we aim to understand the studied phenomena better through model explanations, examining the differences between general and specific domain models. Then, we may evaluate the feasibility of funding these models. Finally, we will include other modalities, like the songs' lyrics, and perform similar studies.

7. Acknowledgments

This work was financed by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code PROEX-12049601/D, Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) – Finance Code 2020/07911-2.

References

- [1] YV Srinivasa Murthy and Shashidhar G Koolagudi. Content-based music information retrieval (cb-mir) and its applications toward the music industry: A review. *ACM Computing Surveys*, 51(3):1–46, 2018.
- [2] Geoffroy Peeters. The deep learning revolution in mir: The pros and cons, the needs and the challenges. In *International Symposium on Perception, Representations, Image*,

Sound, Music, pages 3–30. Springer International Publishing, 2021.

- [3] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pre-trained convolutional network. In *International Society for Music Information Retrieval Conference*, pages 669–674. University of Miami, 2011.
- [4] Jordi Pons and Xavier Serra. musicnn: Pre-trained convolutional neural networks for music audio tagging. *arXiv preprint arXiv:1909.06654*, 2019.
- [5] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Transfer learning for music classification and regression tasks. In *International Society for Music Information Retrieval Conference*, 2017.
- [6] Jeong Choi, Jongpil Lee, Jiyoung Park, and Juhan Nam. Zero-shot learning for audio-based music classification and tagging. In *International Society for Music Information Retrieval Conference*, 2019.
- [7] Hemanth Venkateswara, Shayok Chakraborty, and Sethuraman Panchanathan. Deep-learning systems for domain adaptation in computer vision: Learning transferable feature representations. *IEEE Signal Processing Magazine*, 34(6):117–129, 2017.
- [8] Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual bert? In *Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, 2019.
- [9] Minz Won, Andres Ferraro, Dmitry Bogdanov, and Xavier Serra. Evaluation of cnn-based automatic music tagging models. In *Sound and Music Computing Conference*, 2020.
- [10] Naresh N Vempala and Frank A Russo. Modeling music emotion judgments using machine learning methods. *Frontiers in psychology*, 8:2239, 2018.
- [11] Bob L Sturm, Oded Ben-Tal, Úna Monaghan, Nick Collins, Dorien Herremans, Elaine Chew, Gaëtan Hadjeres, Emmanuel Deruty, and François Pachet. Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, 48(1):36–55, 2019.
- [12] James Bergstra, Norman Casagrande, Dumitru Erhan, Douglas Eck, and Balázs Kégl. Aggregate features and a data boost for music classification. *Machine learning*, 65(2-3):473–484, 2006.
- [13] Jingxian Li, Lixin Han, Xiaoshuang Li, Jun Zhu, Baohua Yuan, and Zhinan Gou. An evaluation of deep neural network models for music classification using spectrograms. *Multimedia Tools and Applications*, pages 1–27, 2021.
- [14] Douglas Eck, Thierry Bertin-Mahieux, and Paul Lamere. Autotagging music using supervised machine learning. In *International Conference of Students of Systematic Musicology*, pages 367–368, 2007.
- [15] Vincent Lostanlen, Joakim Andén, and Mathieu Lagrange. Extended playing techniques: the next milestone in musical instrument recognition. In *International Conference on Digital Libraries for Musicology*, pages 1–10, 2018.
- [16] Xiao Hu, Kahyun Choi, and J Stephen Downie. A framework for evaluating multimodal music mood classification. *Journal of the Association for Information Science and Technology*, 68(2):273–285, 2017.
- [17] Renato Panda, Ricardo Malheiro, and Rui Pedro Paiva. Novel audio features for music emotion recognition. *IEEE Transactions on Affective Computing*, 11(4):614–626, 2018.
- [18] Johan Pauwels, Ken O’Hanlon, Emilia Gómez, Mark Sandler, et al. 20 years of automatic chord recognition from audio. In *International Society for Music Information Retrieval Conference*, 2019.
- [19] Carlos Soares Araujo, Marco Cristo, and Rafael Giusti. Predicting music popularity on streaming platforms. In *Brazilian Symposium on Computer Music*, pages 141–148. SBC, 2019.
- [20] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE transactions on multimedia*, 13(2):303–319, 2010.
- [21] Martin McKinney and Jeroen Breebaart. Features for audio and music classification. In *International Society for Music Information Retrieval Conference*. Johns Hopkins University, 2003.
- [22] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.
- [23] Kristopher West and Stephen Cox. Features and classifiers for the automatic classification of musical audio signals. In *International Society for Music Information Retrieval Conference*, 2004.
- [24] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.
- [25] Keunwoo Choi, György Fazekas, Mark Sandler, and Kyunghyun Cho. Convolutional recurrent neural networks for music classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2392–2396. IEEE, 2017.
- [26] Jordi Pons, Oriol Nieto, Matthew Prockup, Erik Schmidt, Andreas Ehmann, and Xavier Serra. End-to-end learning for music audio tagging at scale. In *International Society for Music Information Retrieval Conference*, 2017.
- [27] Juhan Nam, Keunwoo Choi, Jongpil Lee, Szu-Yu Chou, and Yi-Hsuan Yang. Deep learning for audio-based music classification and tagging: Teaching computers to distinguish rock from bach. *IEEE signal processing magazine*, 36(1):41–51, 2018.
- [28] Yang Yu, Sen Luo, Shenglan Liu, Hong Qiao, Yang Liu, and Lin Feng. Deep attention based music genre classification. *Neurocomputing*, 372:84–91, 2020.
- [29] Deepanway Ghosal and Maheshkumar H Kolekar. Music genre recognition using deep neural networks and transfer learning. In *Interspeech*, pages 2087–2091, 2018.
- [30] Beici Liang and Minwei Gu. Music genre classification using transfer learning. In *IEEE Conference on Multimedia Information Processing and Retrieval*, pages 392–393. IEEE, 2020.
- [31] Yilun Zhao and Jia Guo. Musicoder: A universal music-acoustic encoder based on transformer. In *International Conference on Multimedia Modeling*, pages 417–429. Springer, 2021.
- [32] Ho-Hsiang Wu, Chieh-Chi Kao, Qingming Tang, Ming Sun, Brian McFee, Juan Pablo Bello, and Chao Wang. Multi-task self-supervised pre-training for music classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 556–560. IEEE, 2021.
- [33] Janne Spijkervet and John Ashley Burgoyne. Contrastive learning of musical representations. *arXiv preprint arXiv:2103.09410*, 2021.

- [34] Júlia Luiza Conceição, Rosiane de Freitas, Bruno Gadelha, João Gustavo Kienen, Sérgio Anders, and Brendo Cavalcante. Applying supervised learning techniques to brazilian music genre classification. In *2020 XLVI Latin American Computing Conference*, pages 102–107. IEEE, 2020.
- [35] Carlos N Silla Jr, Alessandro L Koerich, and Celso AA Kaestner. The Latin Music Database. In *International Conference on Music Information Retrieval*, pages 451–456, 2008.
- [36] Jefferson Martins de Sousa, Eanes Torres Pereira, and Luciana Ribeiro Veloso. A robust music genre classification approach for global and regional music datasets evaluation. In *IEEE International Conference on Digital Signal Processing*, pages 109–113. IEEE, 2016.
- [37] Jeronimo Barbosa, Cory McKay, and Ichiro Fujinaga. Evaluating automated classification techniques for folk music genres from the Brazilian Northeast. In *Brazilian Symposium on Computer Music*, 2015.
- [38] Rodrigo Borges and Marcelo Queiroz. Evolution of timbre diversity in a dataset of brazilian popular music: 1950–2000. In *International Conference of Students of Systematic Musicology*, 2018.
- [39] Lucas Maia, Magdalena Fuentes, Luiz Biscainho, Martín Rocamora, and Slim Essid. SAMBASET: A dataset of historical samba de enredo recordings for computational music analysis. In *International Society for Music Information Retrieval Conference*, 2019.
- [40] Felipe Falcão, Nazareno Andrade, Flávio Figueiredo, Diego Silva, and Fabio Morais. Measuring disruption in song similarity networks. In *International Society for Music Information Retrieval Conference*, 2020.
- [41] Angelo Cesar Mendes da Silva, Diego Furtado Silva, and Ricardo Marcondes Marcacini. 4mula: A multitask, multimodal, and multilingual dataset of music lyrics and audio features. In *Brazilian Symposium on Multimedia and the Web*, pages 145–148, 2020.
- [42] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.
- [43] Phillip Rust, Jonas Pfeiffer, Ivan Vulić, Sebastian Ruder, and Iryna Gurevych. How good is your tokenizer? on the monolingual performance of multilingual language models. *arXiv preprint arXiv:2012.15613*, 2020.
- [44] Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234, 2020.
- [45] Zhuang Liu, Degen Huang, Kaiyu Huang, Zhuang Li, and Jun Zhao. Finbert: A pre-trained financial language representation model for financial text mining. In *International Joint Conference on Artificial Intelligence*, pages 4513–4519, 2020.
- [46] Iz Beltagy, Kyle Lo, and Arman Cohan. Scibert: A pre-trained language model for scientific text. In *Conference on Empirical Methods in Natural Language Processing and the International Joint Conference on Natural Language Processing*, pages 3615–3620, 2019.
- [47] Jordi Pons, Thomas Lidy, and Xavier Serra. Experimenting with musically motivated convolutional neural networks. In *International Workshop on Content-Based Multimedia Indexing*, pages 1–6. IEEE, 2016.

Appendix

Here, we present all the confusion matrices obtained in our experiments. We note that the values on these matrices were standardized, so all cells’ sum equals one.

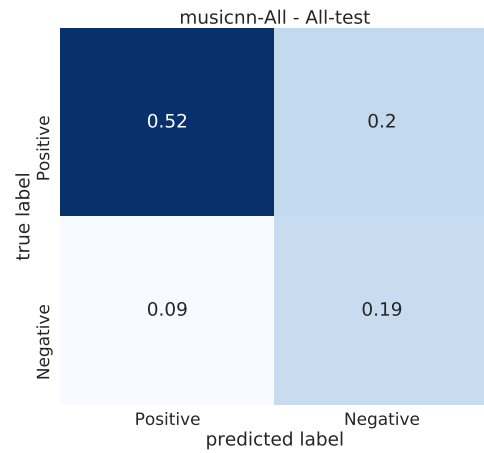


Figure 5: Confusion matrix of musicnn-All on All-test

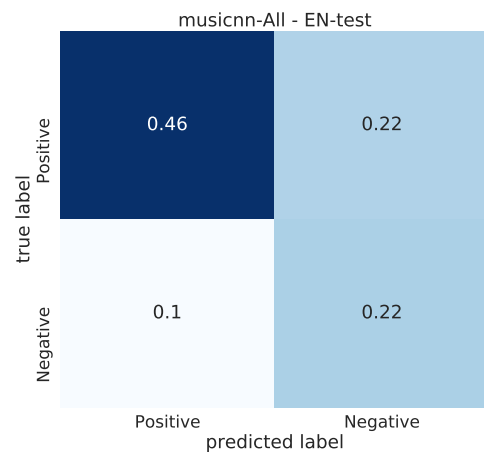


Figure 6: Confusion matrix of musicnn-All on EN-test

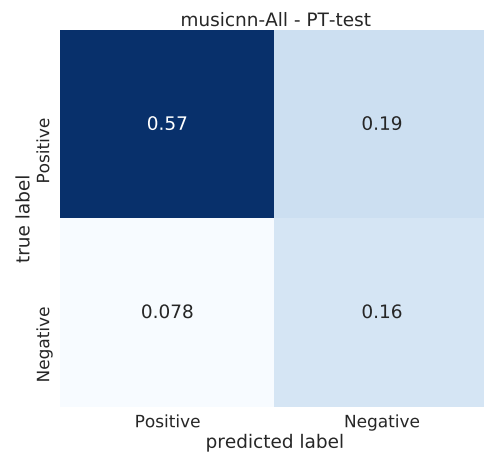


Figure 7: Confusion matrix of musicnn-All on PT-test

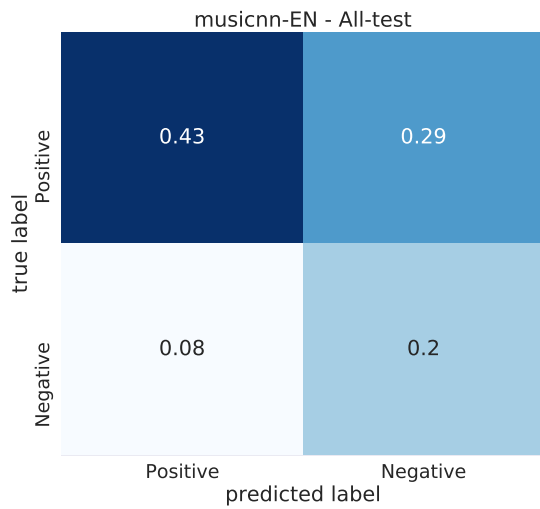


Figure 8: Confusion matrix of musicnn-EN on All-test

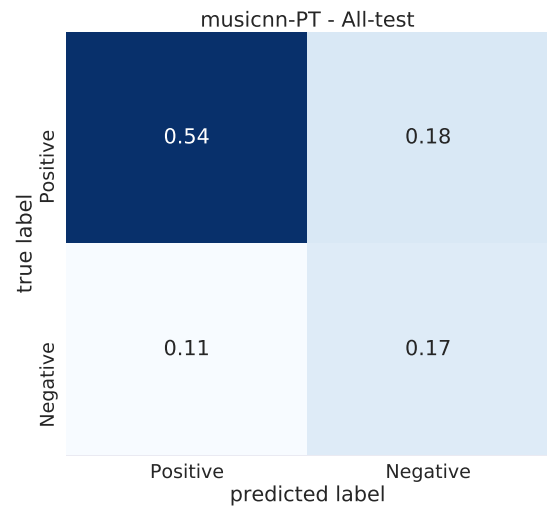


Figure 11: Confusion matrix of musicnn-PT on All-test

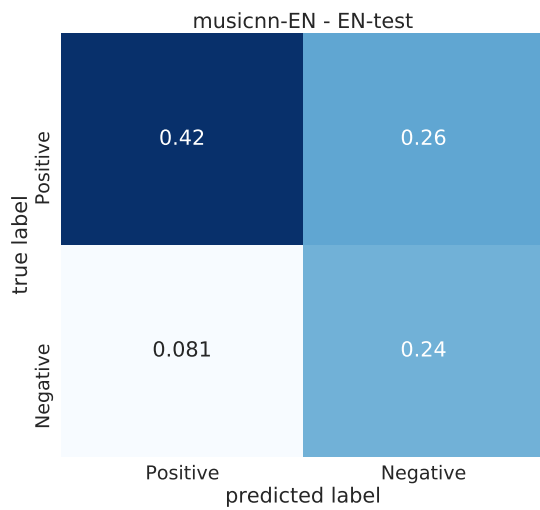


Figure 9: Confusion matrix of musicnn-EN on EN-test

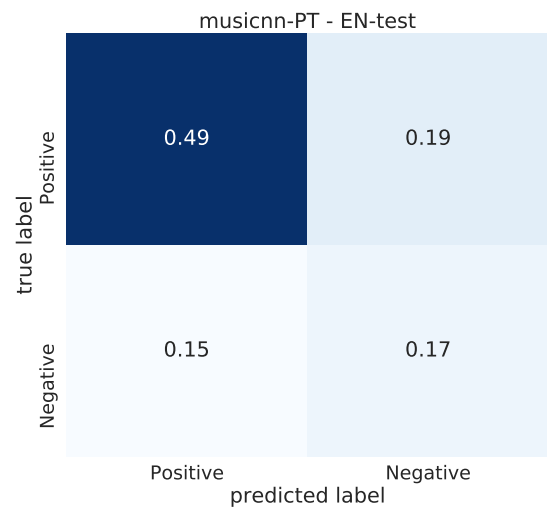


Figure 12: Confusion matrix of musicnn-PT on EN-test

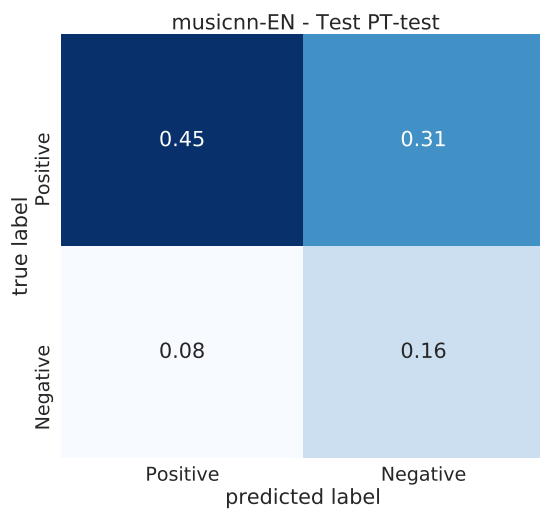


Figure 10: Confusion matrix of musicnn-EN on Test PT-test

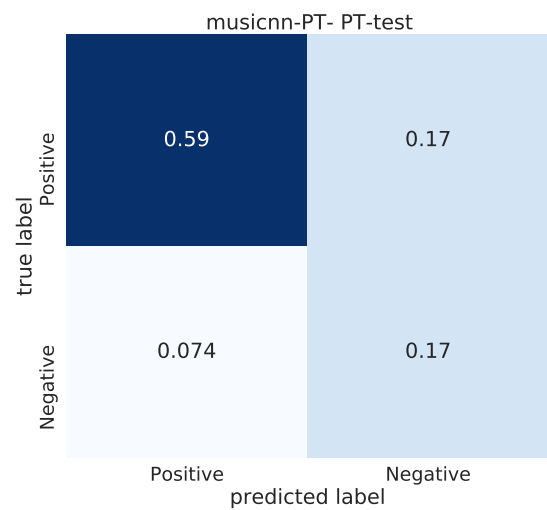


Figure 13: Confusion matrix of musicnn-PT - PT-test

Applications of FFT for timbral characterization in woodwind instruments

Yubiry Gonzalez¹, Ronaldo C. Prati¹

¹Center 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, ronaldo.prati@ufabc.edu.br

Abstract

The conceptualization of the musical timbre, which allows its quantitative evaluation in an audio record, is still an open-ended issue. This paper presents a set of dimensionless descriptors to assess the musical timbre of woodwind instruments in recordings of the fourth octave of the tempered musical scale. These descriptors are calculated from the Fast Fourier Transform (FFT) spectra using the Python Programming Language, specifically the SciPy library. The characteristic spectral signature of the clarinet, bassoon, transverse flute, and oboe are obtained in the fourth musical octave, observing the presence of degeneration for some musical sounds, that is, two given different aerophones may present the same harmonics. It is concluded that the proposed descriptors are sufficient to differentiate the aerophones studied, allowing their recognition, even in the case that there present the same set of harmonic frequencies.

1. Introduction

To characterize sound in general, and musical sound in particular, it is necessary to know the attributes of pitch, intensity, duration, and timbre. The first three characteristics of sound correspond respectively to the acoustic magnitudes of frequency, intensity and time, and they are direct, measurable magnitudes. Timbre, however, is a multi-dimensional and briefly defined attribute that makes it possible to distinguish between different sounds even when they have the same intensity, duration, and pitch. It allows discriminating sounds of different musical instruments even when it is the same musical note, with the same duration and intensity.

The problem of the univocal characterization of musical timbre raises the need to develop descriptors (derived magnitudes, coefficients or functional) that evaluate timbre from digital audio records [1]. Since musical timbre is a phenomenon of auditory perception, many of the investigations are developed in the line of psychoacoustics to evaluate verbal descriptors that reveal measurable attributes of musical timbre [2,3].

Although the psychoacoustic perception of musical timbre cannot be ignored, it must be recognized

that the main timbral characteristics must be, in some way, inscribed within its spectral content [4]. Assume that an FFT calculated from a musical audio record does not contain some significant timbral characteristics. In that case, the deconvolved audio (reverse convolution) could not be distinguished timbrally. However, as we can differentiate between musical instruments after the deconvolution, the initial assumption is not valid.

Therefore, the FFT contains all the essential tonal characteristics. Other research focuses on the presentation of an exhaustive collection of acoustic descriptors in the form of coefficients or functional (Timbre ToolBox, Librosa) that can be computationally extracted from the statistical analysis in the digitization of the spectrum (FFT) and the digital signal itself (spectrogram). These coefficients usually refer to the statistical and mathematical characterization of the maxima in the FFT and the spectrograms, such as the mean value in frequency (centroid) and amplitude, standard deviation, kurtosis, roots or poles of the distribution, arithmetic sequences and geometric in frequency, mean and mean square values of the amplitudes among others [5-9].

There is no consensus on which and how many are the acoustic descriptors' for musical timbre. However, many of them are derivatives or combinations of others and thus highly correlated [5]. It is worth asking which is the minimum set of timbre descriptors that differentiate the musical instruments in a given audio recording.

Despite recent advances in verbal descriptors (psychoacoustic) such as statistical or mathematical descriptors (Timbre Toolbox [5], Librosa [9,10]) of musical timbre, it is still far from being characterized, and it constitutes a challenge for the automated identification of musical instruments.

An alternative approach is to construct the timbral descriptors from that of musical acoustics, using an analogy to the analysis and digital processing of spectra in other areas of knowledge. The FFT represents the two-dimensional plane of the frequencies and intensities (alternatively energy-frequency) present in a signal.

This paper aims to propose a set of acoustically motivated timbral descriptors (mathematical functional or oversized coefficients) that allow the computational

extraction of timbral information, for the automated identification of woodwind musical instruments. For this, a preliminary study of the timbral characteristics in digital musical sounds is presented; using the FFT in the 4th tempered musical Octave of a sample of woodwind instruments: Clarinet, Bassoon, Transverse Flute and Oboe.

In Section 2, the methodology for obtaining the spectrum and their quantification of the Audio records are presented. In Section 3, the coefficients or descriptors and their valuation for the selected aerophones are shown. In Section 4, the chromatogram for the 4th octave of the tempered musical scale is evaluated, and the spectral signature (“fingerprint”) that characterizes each studied aerophone is obtained. Finally, the conclusions and future work are presented in the last section.

2. Methodology

The audio records of the aerophones were obtained from the TinySOL open-source sound library [11], which contains recordings of individual sounds, played in an ordinary way, at a level of dynamic/intensity mezzo forte in a WAV audio format that minimizes information losses, sampled at 44.1 kHz in single-channel (mono) at 16-bit depth. From this library, we have restricted our analysis to only some instruments of the woodwind family of a typical western symphonic music orchestra: Clarinet, Bassoon, Transverse Flute and Oboe.

On these audios, the module for calculating the FFT available in the SciPy library in Python [12] was used to obtain the frequency spectrum; then for the identification of the local maxima, the `find_peaks` function in `Scipy.signal` was used.

Next, the tables of the maximum frequencies, expressed in Hz, with relative amplitudes, normalized considering the maximum amplitude value achieved in each spectrum of the FFT, were obtained in order to compare the various audio spectra with different power levels (Watt) or relative power (dB).

The calculation of the coefficients was performed on the frequency spectrum of the fourth octave of the tempered musical scale (~261 Hz to ~493 Hz) corresponding to musical sounds between C4 to B4 in musical notation. This octave was initially considered as it is the lowest common octave in the tessitura of the selected aerophones; therefore, it is the one that would present the most significant amount of harmonics for each instrument and musical sound.

3. Timbral Coefficients

Musical frequencies make up a set of only 12 different values in the 4th musical octave. Therefore, in a first approximation, musical timbre can be characterized by a limited set of dimensionless quantities related to frequencies and amplitudes in the spectrum. They are related to the fundamental frequency and the partials that

arise from the FFT. In the timbre analysis, the centroid is usually used; its use does not allow a specific acoustic interpretation since the frequency does not correspond to any of the partials frequencies of the sounds studied. The centroid is defined as the average frequency. Where f_i is the harmonic frequencies for each spectrum.

$$\bar{f} = \frac{\sum_{i=1}^N a_i f_i}{\sum_{i=1}^N a_i} \quad (1)$$

In Figure 1, the FFT spectra of the musical sound E4 are shown (329.6 Hz). Note that the centroids do not correspond to any audible signal or related to the musical note E4.

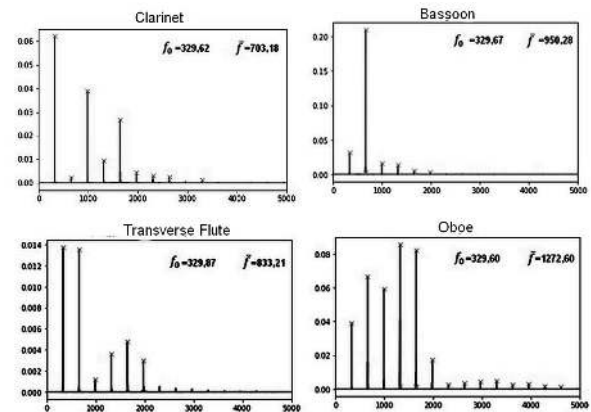


Figure 1: FFT spectra of the musical sound E4 for the studied aerophones, nominal frequency 329.6 Hz.

For all aerophones studied, Figure 2 shows that, in general, the centroid presents variations of less than 15% with respect to its mean value of the 4th octave. Also, Figure 2 shows that, in some sounds, the centroid values are similar for several instruments. This similarity does not allow their differentiation w.r.t. the type of instrument.

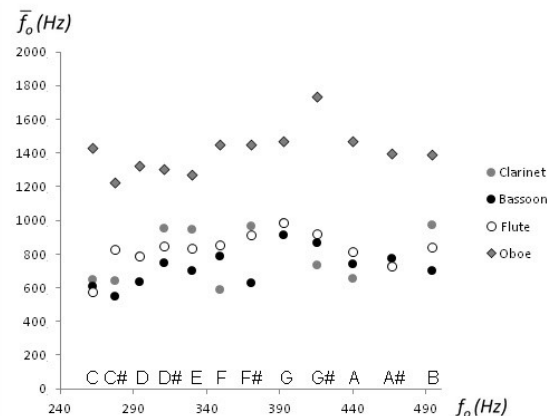


Figure 2: Centroid in 4th Octave of the musical scale

The fundamental frequency f_0 would only coincide with the Centroid $\bar{f}$ if there are no other partial frequencies. This fact is impossible in real musical instruments due to the harmonics, the superposition and the beats inside the resonator tubes of the aerophones. The separation between the fundamental frequency and the mean value of the frequencies would be important in musical acoustics. The Affinity coefficient describes how far the spectrum is from the ideal case, that is, how far the maximum Principal f_0 is from the mean value in frequency or Centroid ($\bar{f}$)

$$A \equiv \frac{\bar{f}}{f_0} \quad (2)$$

The affinity coefficient allows discriminating different sounds with the same Centroid (figure 3). Figure 4 shows the values of A for the 4th octave.

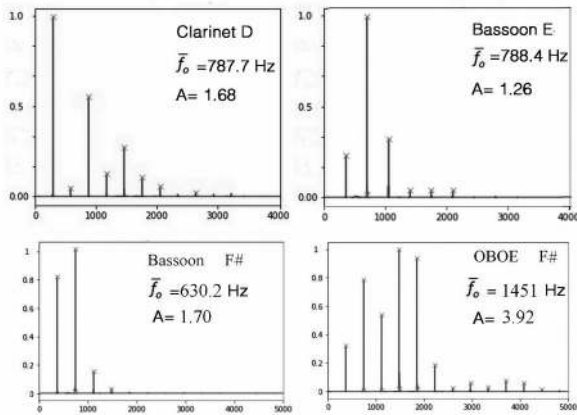


Figure 3: Fourier spectrum with different coefficients. Top: Examples with very similar centroids. Bottom: Examples with Equal musical sounds.

It can be observed in Figure 1, that different musical instruments have different amplitudes in their fundamental frequency. This motivates another descriptor: the Sharpness coefficient (S).

$$S \equiv \frac{a_0}{\sum_{i=1}^N a_i} \quad (3)$$

Ideally, $S=1$ in a “pure” sound that would have a single maximum without secondary frequencies. By construction, $S \leq 1$. Figure 5 exemplifies Sharpness in different musical instruments and for the same musical instrument with different sounds.

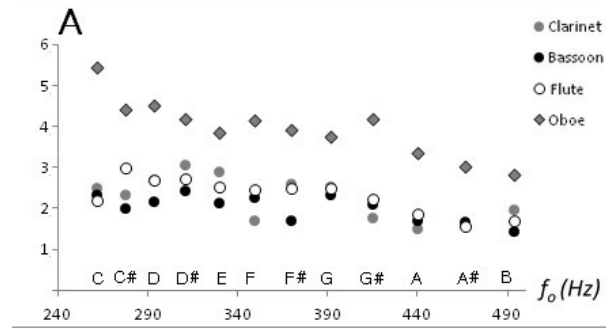


Figure 4: The values of affinity (A) for the 4th octave

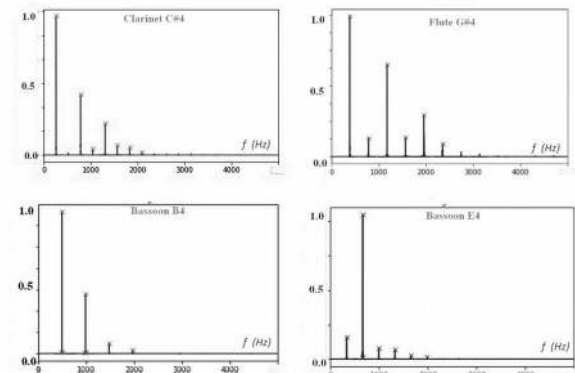


Figure 5: Timbral coefficient S . Above: The FFTs of the Clarinet in C# (277.2 Hz) and of the Transverse Flute in G# (415.3 Hz) both with $S = 0.55$. Bottom: Bassoon FFT in B4 and E4, $S = 0.98$ and $S = 0.11$ respectively

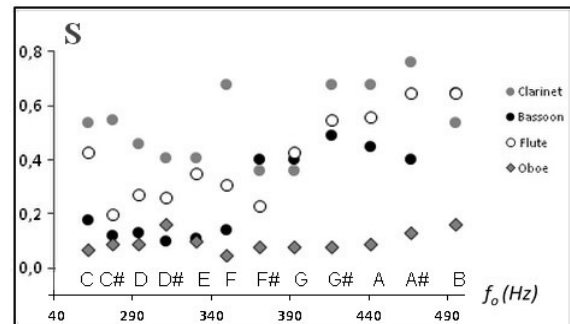


Figure 6. Sharpness (S) in the 4th octave.

Figure 6 shows the variation of the Sharpness of the aerophones in the 4th Octave. It is observed that, in general (except for F#), the clarinet is sharper than the transverse flute, and is sharper than the bassoon. In any case, the Oboe is the least sharper of all. In other words, musical sounds have better sharpness on the clarinet and worse sharpness on the Oboe.

4. Spectral Signature

In general, the identification of musical instruments and of the wave emitting source can be made through the analysis of their spectrum. In the case of

electromagnetic radiation, the specific profile of intensities versus wavelength (absorption or emission) allows the identification of the emitter (is it a gas, a reflective surface or an absorbing medium) and is called Spectral Signature. By analogy, we understand as "Spectral signature of a musical instrument", the distribution of frequencies in the Fourier Transform Spectrum of musical sounds (tempered scale) that allow their identification.

Suppose we restrict ourselves to the aerophones studied, which are only a small sample of all common musical instruments. In that case, a discriminatory identification between them can be made from the spectra of the presented FFT spectra for the 4th octave records. (Figure 7).

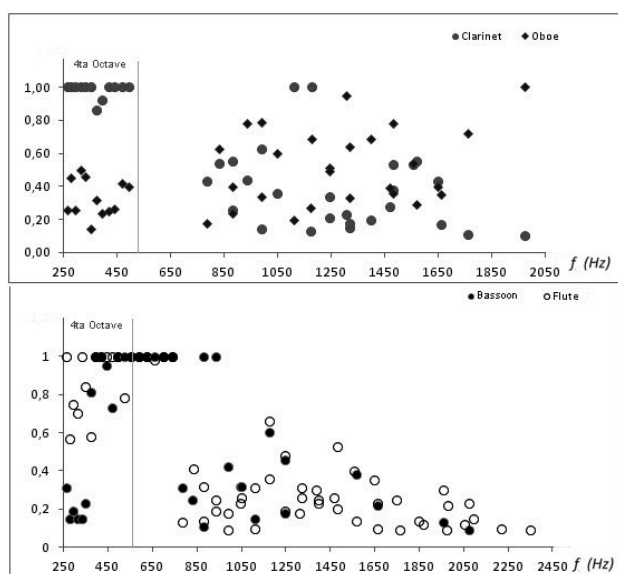


Figure 7. Spectral Signatures in the 4th Octave for Aerophones: Top: Oboe and Clarinet. Bottom: Bassoon and Transverse Flute.

If all the harmonics of a certain musical sound are present, this would allow, to discriminate the sounds between each instrument (Table 1). The harmonic frequencies (f_n) are related to the fundamental frequency through the integer multiplicity ($n = 2, 3, 4, \dots$):

$$f_n = n f_0 \quad (4)$$

This discrimination, regarding which harmonics are present or absent, is not univocal. It presents degeneration, in the sense that for some musical sounds, two different aerophones present the same harmonics (highlighted with the symbol * in table 1).

Musical sound and frequency (Hz)	Characteristic Harmonic Frequencies			
	Clarinet	Bassoon	Flute	Oboe
C4 : 261,6	{3,...,8}	{2,...,8}	{2,...,5}	{2,...,9}
C4#: 277,2	{3,...,7}	{2,...,6}	{2,...,8}	{2,...,9}
D4 : 293,7	{2,...,7,9}	{2,...,7}*	{2,...,7}*	{2,...,8,10,...,15}
D4#: 311,1	{2,...,10}	{2,...,7}*	{2,...,7}*	{2,...,12}
E4 : 329,6	{2,...,9}	{2,...,6}*	{2,...,6}*	{2,...,13}
F4 : 329,67	{2,...,5}	{2,...,6}*	{2,...,6}*	{2,...,13}
F4#: 370	{3,...,7}	{2,...,4}	{2,...,5}	{2,...,12}
G4 : 494	{2,...,5}	{2,...,5,8}	{2,...,6}	{2,...,12}
G4#: 415,3	{2,...,8}	{2,...,6}	{3,...,6}	{2,...,13}
A4 : 440	{2,...,5}	{2,...,7}	{2,...,6}	{2,...,11}
A4#: 466,2	{2,...,8}	{2,...,6}	{2,...,5}	{2,...,9}
B4 : 494	{2,...,6}*	{2,...,5}	{2,...,6}*	{2,...,9}

Table 1: Harmonic set as a function the sound and musical instrument of the fourth octave.

(*) Denotes degeneracy: two different instruments with the same set of harmonics.

However, as the FFTs are not the same between any two musical instruments, the different relative intensities, through the Affinity and Sharpness coefficients, would allow completing of the spectral signature and the univocal identification of the instrument (Figure 8).

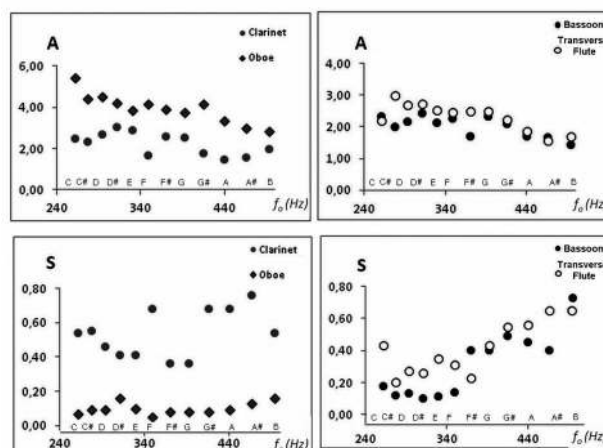


Figure 8. Timbral Affinity and Sharpness

The variation of these coefficients for musical sounds in the 4th octave, allows the spectral signature to be uniquely identified without degeneration.

On the other hand, in a given spectrum, there are always partial or secondary frequencies to f_0 , that can be counted. However, it can also happen that a secondary frequency is not strictly an integer multiple of the fundamental frequency but whose frequency is very close to an integer multiple of f_0 .

In a way, the sound would be more "harmonic" than if its frequency value were very different from the multiplicity of the fundamental frequency. We have

defined the coefficients from acoustics' perspective where an non-oscillating oscillator in harmonic motion is known as an anharmonic oscillator. The difference is more than mere semantics.

This was made by design as Inharmonic coefficient defined by Peters [14] quantifies only the frequencies that are multiples of the fundamental (integer k values). However, in our definition of the Harmonicity coefficient, all frequencies of the Fourier spectrum are quantified. To describe this property, the timbral coefficient of Harmonicity (H) is proposed and defined as:

$$H \equiv \sum_1^N \left(\frac{f_j}{f_0} - \left[\frac{f_j}{f_0} \right] \right) \quad (5)$$

Where the $[]$ denotes the integer part.

This timbral coefficient evaluates how harmonic the partial or secondary frequencies ($f_1, f_2, f_3... f_j$) in the FFT spectrum. The idea is that any frequency f is a harmonic of f_0 if the quotient between them is an integer. Figure 9 shows this variation in examples of FFT spectra with different harmonicity.

Our definition is very different from the inharmonicity descriptors [14,15]. First of all, we do not weigh the amplitudes, for the purposes of multiplicity in frequency. Secondly, note that if f_k is not an integer multiple of f_0 , it will not appear in the summation nor will it be considered in Bullock's expression [15], but it will influence the harmony of the set and, therefore, the expression shown in Equation (5).

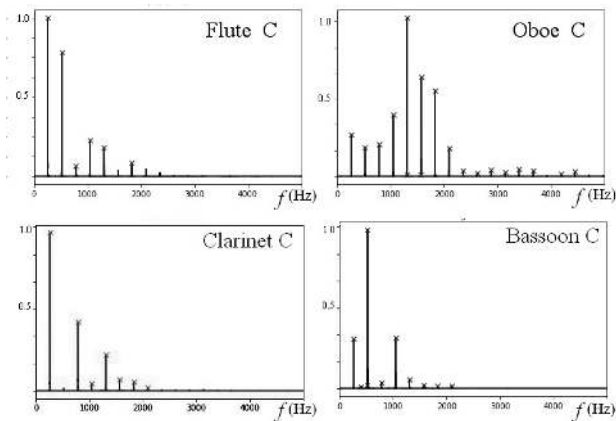


Figure 9. Comparison of partial or secondary frequencies in the sounds C, between the Transverse Flute ($H= 0$), Oboe ($H = 16.9$), Clarinet ($H= 2.00$) and Bassoon ($H= 7.48$).

Figure 9 shows a comparison of secondary frequencies in the sound C4 for the instruments we are analysing. Analysing the figure, we observe that the oboe is more inharmonic than the flute, with harmonicity coefficients of $H = 0$ and $H = 16.9$, respectively. Comparing it with the coefficient proposed by Peters [14], the same value is obtained for both instruments (Zero) and, as a consequence, it does not describe the harmonicity that we define.

The C4 sound of the Flute is much more harmonic than the Oboe, even though the latter has many more secondary frequencies, some of which can be harmonics of f_0 , as is the case of the second, third and fourth maximum. Furthermore, if the secondary frequencies are all harmonics of f_0 then $H=0$. Every time there is one or more frequencies that are not harmonics of f_0 , the j -th term, in the sum, will be non-zero, and H increases. The maximums of the spectrum distribution of the Flute in Figure 9 are all integer multiples of the fundamental and therefore it is highly harmonic ($H = 0$).

For the fourth octave, the results of the harmonicity assessment are shown in figure 10. The Transverse Flute has the highest harmonicity (H near zero), that is, its secondary frequencies are usually integer multiples of the corresponding fundamental sound, while the Oboe is the most anharmonic (large H). We note that the quality of harmonicity varies with the musical note and instrument, so for the characteristic sound of reference A= 440 Hz the Bassoon and Clarinet are completely harmonic, and the Transverse Flute is completely harmonic also for C, D, E, and A#.

The Harmonicity coefficient (H) allows, through the FFT, to quantify the number of harmonics present for a given fundamental frequency. Therefore, in a first approximation, we can affirm that it is related to "color" or "sonority" of these instruments [13].

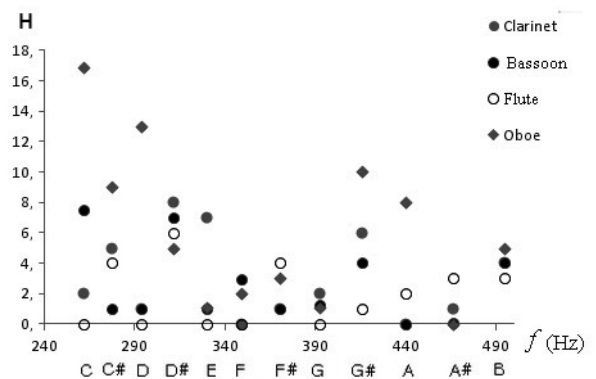


Figure 10. Harmonicity (H) in the 4th Octave.

6. Conclusions

The Fast Fourier Transform (FFT) allows analyzing digital audio records on the tempered musical scale, generating a unique spectrum that characterizes it. As a finite, bounded and countable collection of discrete frequencies with different amplitudes, this spectrum

provides two-dimensional arrangements of frequency and amplitudes for the musical sounds of aerophones. Moreover, they can be analyzed using the usual automated signal analysis techniques. Therefore, the characteristics of the audio records, including the timbre, are susceptible to being automated by artificial intelligence techniques.

It is possible to calculate oversized coefficients that reflect timbral characteristics of the fundamental frequency and harmonic distribution in the FFTs of simple 4th-octave monophonic music registers for woodwind instruments. Thus, the proposed timbral descriptors, Affinity and Sharpness, allow distinguishing the timbre of Clarinet, Bassoon, Transverse Flute, and Oboe.

The Spectral Signature of the wooden aerophones, which allows the univocal identification of them, can be obtained through the FFT by the distribution of harmonics present outside the loudness range of the 4th octave. The number of harmonics present and their succession does not always characterize the timbre between different aerophones. There may be degeneration (two or more instruments with equal harmonics). However, the proposed coefficients allow a unique timbral identification in each musical sound studied.

We conclude that the proposed descriptors allow describing timbral characteristics of the instruments studied, in addition to differentiating them, allowing their recognition from the FFTs.

Future work will implement the FFT techniques and the described timbric coefficients until completing the tessitura of the selected aerophones. Later the study will be extended to incorporate the temporal variations of the timbre coefficients in the case of melodic fragments that are important for quantising the timbre in the harmony and the musical composition.

References

[1] M. Adeli, J. Rouat, S. Wood, S. Molotchnikoff, and E. Plourde, A flexible bio-inspired hierarchical model for analyzing musical timbre. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(5): 875-889. 2016.

[2] W. Jiang, J. Liu, X. Zhang, S. Wang and Y. Jiang, Analysis and Modeling of Timbre Perception Features in Musical Sounds. *Applied Sciences*, 10 (3): 789. 2020.

[3] D. Fourer, Jean-Luc Rouas, P. Hanna, M. Robine. Automatic Timbre Classification of Ethnomusicological Audio Recordings. *International Society for Music Information Retrieval Conference (ISMIR 2014)*, Oct 2014, Taipei, Taiwan.

[4] M. A. Loureiro, and H. B. D. Paula. Timbre de um

instrumento musical: caracterização e representação. *Per Musi*, Vol 14, pp. 57-81. 2006.

[5] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *Journal of the Acoustical Society of America*, 130 (5): 2902-2916. 2011.

[6] J. Garcia and G. Tzanetakis,. Musical Instrument Classification Using Individual Partial in *IEEE Transactions on Audio Speech and Language Processing* · February 2011.

[7] S. Joshi and A. Chitre. Identification of Indian musical instruments by feature analysis with different classifiers. *Proceedings of the Sixth International Conference on Computer and Communication Technology*, pp. 110-114, 2015.

[8] H. Ezzaidi, M. Bahoura and G. E. Hall. Towards a characterization of musical timbre based on chroma contours. *International Conference on Advanced Machine Learning Technologies and Applications*, Springer, Berlin, Heidelberg, pp. 162-171. 2012

[9] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs and G. Widmer. Madmom: A new python audio and music signal processing library. In *Proceedings of the 24th ACM international conference on Multimedia*. pp. 1174-1178. 2016.

[10] B. McFee, C. Raffel, D. Liang, D. P. Ellis, M. McVicar, E. Battenberg, and O. Nieto. Librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference Vol. 8*, pp. 18-25. 2015.

[11] C. E. Cella, D. Ghisi, V. Lostanlen, F. Lévy, J., Fineberg, and Y.Maresz,. OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration. *arXiv preprint arXiv:2007.00763*. 2020.

[12] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau and P. van Mulbregt,. *SciPy 1.0: fundamental algorithms for scientific computing in Python*. *Nature methods*, 17(3), 261-272. 2020.

[13] Y. González and R. C. Prati. Characterization of the sonority associated with woodwinds instruments through spectral analysis. *Conference: Brazilian Symposium on Computer Music: São João Del Rei, Minas Gerais, Brazil, September 2019*.

[14] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. *CUIDADO Ist Project Report*, 54(0), 1-25. 2004.

[15] J. Bullock. Implementing audio feature extraction in live electronic music (Doctoral dissertation, PhD thesis, University of Birmingham). 2008.

Real-time Qualification of Percussive Sounds Based on Correspondences Between Schaeffer's *Solfège* and Low-level Audio Descriptors

Sérgio Freire¹, José Henrique Padovani¹, Caio Campos¹

¹LaPIS – School of Music/Federal University of Minas Gerais
Av. Antônio Carlos, 6627 – 31270-901 Belo Horizonte, MG

sfreire@musica.ufmg.br, jhp@ufmg.br, costacaiocampos@gmail.com

Abstract. Pierre Schaeffer's typomorphology (1966) proposes seven criteria of musical perception for the qualification of sound objects, which form the basis of his musical theory. This *Solfège* fits well into contexts where pitch is not the dominant dimension. Relying on similarities between the practice of reduced listening and the utilization of low-level audio descriptors, we present the first version of a real-time setup in which these descriptors are applied to qualify percussive sounds. The paper describes the tools and strategies used for addressing different criteria: envelope followers with different window sizes and filtering; detection of transients and amplitude modulations; extraction and counting of spectral components; estimation of intrinsic dissonance and spectral distribution; among others. The extracted data is subjected to simple statistical analysis, producing scalar values associated with each segmented object. Finally, we present a variety of examples.

1. Introduction

The association of audio descriptors with the Schaefferian theory is not rare in recent literature. One can find it in analytical contexts [1], in the association of acoustic data and subjective labels [2, 3, 4], and automatic sound indexing [5]. Solomon [6], although not using audio descriptors and not explicitly using this theory, presents a conceptual approach for the classification and association of percussive sounds similar to the one intended here.

In the last years, we developed a real-time setup in Max¹ to qualify percussive sounds using the *Solfège*'s perceptual criteria, aiming at its application in interactive situations. Our approach uses time and frequency domain representations to tackle the diversity of criteria and sounds. In this paper, we present its first version, tested with diverse sound types. The intention was to build a general framework upon which we could later refine some specific tools. The background assumption is that the practice of reduced listening has similarities with the application of low-level audio descriptors.

The text organizes as follows. First, we present the criteria for musical perception that constitute the Schaefferian *Solfège*, followed by the segmentation and pre-processing procedures used in the setup. The description of the implemented tools comes next. Then we discuss the intended correspondences between the criteria and the

descriptors. A section with examples and their discussion completes the presentation of the setup.

2. Schaeffer's Criteria

In the Treatise on Musical Objects (TOM)[7, 8], Pierre Schaeffer introduced a theoretical framework and a practical methodology for classifying and manipulating electro-acoustic sounds, what he would come to call *Musical Research Program*. By inverting the traditional notion of *solfège* – associated with the practice of singing intervals and scalar excerpts through the solmization of musical notes – Schaeffer proposed a “generalized *solfège*”, based on the “art of better hearing”.

To furnish means to his proposal of describing perceptual aspects of sound phenomena — despite any referential or causal events such as physical or instrumental factors that may have generated them — Schaeffer proposes, borrowing the Husserlian concept of *epoché*, a *reduced listening*. By this term, the author refers to a conscious attempt to scrutinize the attributes of sound objects by taking into account the perceptual dimensions of pitches, durations, and intensities. To enable a more detailed description of sound objects in these three dimensions, Schaeffer proposes seven typomorphological criteria: *mass*, *dynamic*, *harmonic timbre*, *melodic profile*, *mass profile*, *grain*, and *allure*.

The seven typomorphological criteria proposed by Schaeffer allow him to build a methodological framework to typify, qualify, and evaluate audible characteristics of sound objects with the help of categories like *types*, *classes*, *genres*, and *species*. This comprehensive method is compiled in the TARSOM², a Summary Diagram that outlines the analytical concepts developed by the author. Thus, while the first three columns of this table – *types*, *classes*, and *genres* – relate the criteria (7 rows), respectively, to *typology*, *morphology*, and *characterology*, the following six seek to draw connections between these seven morphological criteria and the perceptual dimensions of *pitches*, *intensities*, and *durations*, employing the *site/calibre* binomial (two columns per perceptive field).

As Michel Chion clarifies (section 25, on the perceptual field, in [9]), each criterion has a more evident relationship with one or more of these perceptual fields. In

¹www.cycling74.com

²*Tableau Récapitulatif du Solfège des Objets Musicaux*, presented at the pp. 584-587 of the original edition[7] and on pp. 464-467 of the English version[8] of the TOM.

Table 1, we present the summary description of the morphological criteria – provided by Chion in section 88 of the same work, but here described with fragments from the English translation of TOM [8].

3. Pre-processing and Segmentation

Defining the adequate sound portion to be analyzed in a real-time setup is crucial for working with reliable data. In our program, the inputs are audio streams delivered by microphones, pickups, or mixers, presenting diverse background noise and dynamic ranges. It is not difficult to control background noise by observing the input amplitude during a “silent” period. On the other hand, the dynamic range poses some challenges, like the leakage from other sources (including sounds coming from loudspeakers). We decided to set a maximal dynamic range of 40 dB, which can cover the range of the majority of instruments [10] but may cut off earlier some long resonances. In the development phase, we decided to use a set of pre-recorded sounds, which offers not only variety but also repeatability, two relevant factors for building and refining tools. The sound selection depicted in Table 8 was based on Schaefferian typology. These sounds become real-time inputs to the setup³, running with a sampling frequency of 48 KHz.

Portions to be analyzed are segmented between onset and offset points. A new segmentation clue may occur before the offset; in these cases, this clue determines the offset of the last event and the onset of the present one, which is qualified as “slurred”. The detection of onsets and offsets relies on the comparison of a dynamic envelope with two thresholds, 6 dB and 3 dB, respectively, above the background noise level. This envelope is an RMS curve estimated with a short window size – 256 points – and a hop size of 64. We will refer to this curve as *rms256:4*, and use a similar notation for other envelopes. The implementation uses [gen~] routines, which employ native audio signal processing and offer more efficiency and precision. A low-pass filter (a single one-pole filter, with a -6 dB per octave attenuation) smoothes these envelopes, and we use different cutoff frequencies depending on the purpose. The detection of onsets and offsets employs a cutoff frequency of 4 Hz. The same setting applies to the detection of the end of an attack (in this case, the input signal may pass through a filter before the estimation of the envelope). Routines dedicated to attack profiles and iterative grains use the same envelope with a cutoff frequency of 30 Hz. The attack profiles are expressed by a control-rate version of this curve. We also use an *rms2048:4* curve, low-pass filtered at 10Hz, for the global dynamic envelope and the detection of allures. Other processes use spectral peaks values estimated by the [sigmund~] object [11], using the same window and hop sizes.

³The sound files used in this study are available in the following link: https://bit.ly/sbcm2021_soundExamples

4. Implemented Tools

4.1. Time Domain Low-level Descriptors

Duration is a simple attribute, whose value is the time interval (in ms) between onset and offset. The estimation of onsets and offsets was described in Section 3. The dynamic profile is the portion of the *rms2048:4* curve comprised between onset and offset. The dynamic level is a simple feature, represented by the mean value and standard deviation of the same curve. Even if we try to improve the correspondence between dynamic levels and perceptual attributes by observing the total duration and spectral region, it is necessary to remember Schaeffer’s words: “For sounds with unremarkable mass and profile this dynamic field is *almost unknown*.” (TOM[8], p. 432)

Schaeffer has stressed the importance of the attack for some sound typologies and the perception of durations. In the case of percussive sounds, this is a primordial characteristic. Therefore, we prefer to analyze the entire attack profile, which may surpass 300 ms, instead of stopping at the point usually called the “end of attack” [12]. For the same reason, this point will be called the “attack first plateau”. We define the “first plateau” as the moment when the derivative of the low-pass filtered audio-rate *rms256:4* curve from the (possibly filtered) input audio stream comes near (or cross) to zero, just after having surpassed a predetermined positive threshold. We call the ratio between the amplitude difference and the time interval that occur between the onset and the first plateau as **FPSlope**. Since we prefer to consider multiple fast strokes (such as flans, drags, ricochets) as belonging to the same profile, a value of 200 ms stays as the reattack threshold. Depending on the settings (filtering and thresholds), the algorithm may not detect soft attacks⁴. On the other hand, some iterative sustainments are considered single allured sound objects, even when the distance between peaks exceeds the given threshold. Figure 1 depicts the attack profiles of eight percussive sounds with varied characteristics.

4.1.1. Allures

As practically any sound event with mechanical origins presents small fluctuations in its envelope, it is necessary to define a minimum threshold for the occurrence of a noticeable allure. In our setup, this threshold is a value expressed in dB (3 dB by default). The detection of allures and related features uses a control-rate version of the *rms2048:4* curve of the input stream, expressed in dBFS. The sign change of the derivative of the signal (crossing a tiny region around zero) is indicative of a possible peak or trough, which may be validated if the difference between the present peak and the last trough (and vice-versa) exceeds the chosen threshold. The estimated descriptors are (1) number of occurrences; (2) mean value and standard deviation of the (a) difference of intensity between peaks and troughs; (b) time interval between successive peaks;

⁴These parameters (reattack time and sharpness) help to redefine the fluids limits between the context (“whether the criteria are artificially put into a structure...”) and the contexture (“...or naturally form a structure”) of percussive sounds in diverse situations. (TOM[8], p. 402)

Table 1: Schaeffer's descriptions of typomorphological criteria and the related perceptual fields

critereon	description	fields
<i>mass</i>	"...quality through which sound installs itself (in a somewhat a priori fashion) in the pitch field." (p. 412)	pitches
<i>dynamic</i>	"...the variation in intensity of this sound in the course of its duration" (p. 33); "...its <i>energetic development</i> " [<i>évolution énergétique</i>] (p. 174)	intensities durations
<i>harmonic timbre</i>	"...more or less diffuse halo and more generally the secondary qualities that seem to be associated with mass and enable us to describe it." (p. 412)	pitches
<i>melodic profile</i>	"Neumes, although intended to represent variations in a specific source (the voice), can provide us with a model. (...) Type of sounds deliberately varied in tessitura" (p. 458)	pitches intensities durations
<i>mass profile</i>	"The <i>mass profile</i> is made up of all the (perceived) intensities of the various components of the spectrum of a sound." (p. 433)	pitches intensities durations
<i>grain</i>	"...a microstructure, generally due to sustainment from a bow, a reed, or even a drum roll. This property of <i>sound matter</i> reminds us of the <i>grain</i> of a textile or a mineral. (...) We find ourselves in a zone where two sensations from the same phenomenon [bassoon reed] merge: the perception of pitch from the beats, and the perception of beats from differentiation of the impacts" (p. 437)	pitches intensities durations
<i>allure</i>	"...the more or less regular oscillations that are its hallmark also cause variations in pitch (vibrato in stringed instruments, singers, etc.) and harmonic timbre. We could say that allure is made up of many factors (...), the most important of which are associated with the dynamic and pitch of sounds." (p. 438)	pitches intensities durations

(c) proportion between peak/trough and trough/peak intervals (symmetry); and (d) maximal value of the derivative in each inflection (spikiness). For a rough estimation of the distribution of peaks through the duration of the entire sound, their temporal centroid and spread are also calculated (in a time series filled with zeros, we insert the value 1 for each detected peak).

4.1.2. Grains

We use two different algorithms for the estimation of the presence and quality of grains in the audio stream, one linked to iteration, the other to resonance and friction types. These last two types are treated jointly under the term tiny grains. For iterative grains, we use the derivative of an *rms256:4* audio-rate curve low-pass filtered at 30 Hz. This signal goes through a Schmitt trigger (with thresholds -0.01 and 0.01). When its value goes below the lower limit, there is an indication of a possible grain, which is confirmed if the time interval between two occurrences has a value below 75 ms. The grain amplitude correlates with the difference between the peaks and troughs of the RMS curve in the same time interval. The estimated descriptors are (1) number of iterative grains; (2) mean value and standard deviation of (a) grain amplitudes; (b) time interval between successive peaks.

The estimation of tiny grains is done on the proper audio stream, or better, on its derivative. We assume that a granular signal will change directions more often than a smooth one. Although it may be objected that signals with

higher frequencies will also bend more often (and with a larger difference between samples, due to the sampling resolution), our experience has showed that in the realm of percussive sounds the granular influence is stronger than the register⁵. An alternative would be the use of the spectral modeling synthesis (SMS), an algorithm proposed by Serra and Smith [13] to separate between deterministic and residual parts of a signal, using a frequency domain representation. However, as it presupposes the existence of a fundamental frequency, it is not adequately apply to most percussive sounds.

4.2. Frequency Domain Low-level Descriptors

This set of descriptors employs the spectral peaks estimation algorithm used by the `[sigmund~]` object, with the following parameters: analysis size of 2048 points; hop size of 512 points; 20 peaks; outputs (peaks, envelope, and pitch). These outputs are also controlled by the onset and offset descriptors described above and have a refresh rate of 10.67 ms with a sampling frequency of 48 kHz. For each frame analysis, we calculate the following descriptors.

Pct50 and **pct80** (number of peaks for energy percentiles 50% and 80%). According to Parseval's theorem, the energy of a time signal equals the sum of the energy of the absolute values of its frequency components. Applying this principle to `[sigmund~]` outputs (peaks and

⁵A few examples: sine waves with frequency of 100 Hz (2 grains), 1000 Hz (21 grains), 10 kHz (213 grains), pink noise (ca. 300 grains) and white noise (ca. 350 grains).

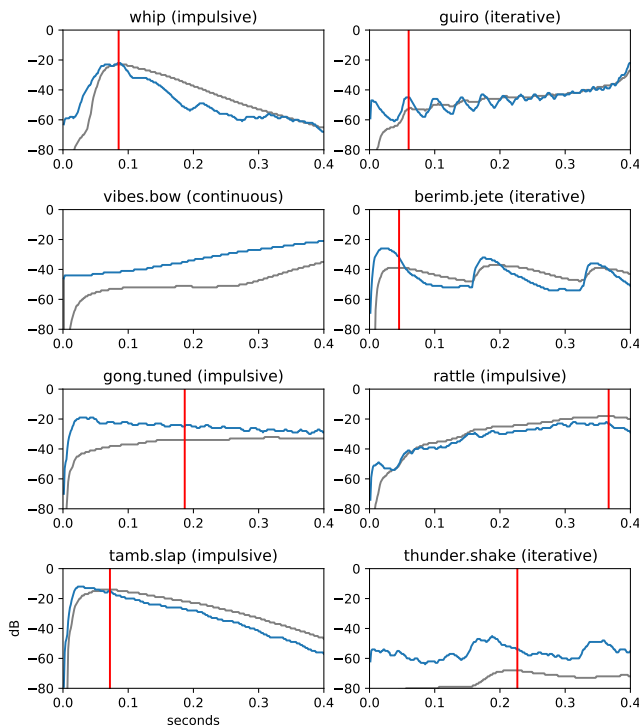


Figure 1: Attack profiles of eight sounds. The blue and gray curves derive from *rms256:4* estimations; the latter is calculated from a filtered audio. The red line indicates the first plateau. The gender of the attack profiles are indicated between parentheses.

envelope), it is possible to estimate the number of peaks needed to obtain the 50% (-3 dB) and the 80% (-1 dB) energy percentiles. The descriptor outputs two curves with the estimated number of peaks. For sounds with a more continuous spectral distribution, 20 peaks may not reach the chosen percentiles. In such cases, the output will be 20 peaks, and further information is obtained through the next descriptor. **20P/total** (percentage of sound energy represented by up to 20 peaks) estimates the amount of energy represented by the set of peaks calculated for each analysis frame and ranges between 0 and 1. The descriptor, along with the last one, helps differentiate sounds between the classes tonic and node. **MPP** (most prominent peak) is a very simple descriptor, represented by the curve formed by the values (expressed in Midicents) of the peaks with the largest amplitude in each analysis frame.

Estimated fundamental frequency. The object `[sigmund~]` outputs a value in Midicents for frames considered to bear a fundamental frequency and the value -1500 for unpitched frames. Our descriptor outputs a scalar (ratio of pitched to total frames—**unpitched/total**), a curve with all numbers, in which -1500 is substituted by 1, another curve with only the pitched values, and a third with 1 for pitched, and 0 for unpitched frames.

Intrinsic dissonance. The estimation of the intrinsic dissonance uses an implementation of the algorithm developed by Sethares [14], using the frequency and am-

plitude values delivered by the sigmund analysis. As he prescribes the use of SPL pressure values, we used 0.00001 for the minimal audible reference.

SC (spectral centroid). We use a `[gen~]` routine delivered with the Max program since its version 6 for the estimation of the spectral centroid. Instead of using a nominal value in Hz, we use values in Midicents, which define a scale ranging from 15.5 to 155 in the audible range.

Δ peaks (interval between the lowest and highest peaks). For each frame, we calculate the difference between the highest and the lowest estimated peaks, which is also expressed in Midicents.

Spectral region. The starting point for the definition of a region is a rough division of the audible spectrum in three ranges. The first three octaves (20–160 Hz) define the low range, the four intermediate octaves (160–2,560 Hz) the medium range, and the last three octaves (2,560–20,000 Hz) the high range. We estimate the energy carried by the peaks in each analysis frame for each of these ranges. If none of these ranges hold 40% or more of the total energy, the sound frame is classified as wide-band, labeled as (7). Otherwise, any range with more than 40% of the total energy contributes to qualifying one of the six spectral combinations: (1) Low, (2) Low/Medium, (3) Medium, (4) Low/High, (5) Medium/High, (6) High.

The flowchart in Figure 2 depicts the main processes involved in the estimation of all low-level descriptors. The time series generated by most descriptors are subjected to simple statistical analysis just after the offset. We adapted the algorithms given in [12], and have chosen the following scalar descriptors: mean value and standard deviation; temporal centroid and spread (normalized by the total duration); skewness; kurtosis; crest; flatness. The correlations with the Schaefferian perceptual attributes rely on these values.

5. Intended Correspondences

As stated in the Introduction, our purpose is to find correlations between low-level descriptors and the criteria of musical perception defined by Schaeffer. In this study, we reduced the seven criteria to six, rearranging them as follows. Since in the realm of percussion (and of everyday sounds) tonic sounds are not the rule, we prefer to unify mass and harmonic timbre under one single category, relying on an observation made by Schaeffer, “considering them rather as connecting vessels, with the exception of certain specific examples...” (TOM, p. 412) Melodic and mass profiles are joined by the same reasons. Although the same set of descriptors supports the qualification of sounds under the four mentioned criteria, we prefer to put the profiles in a dedicated category since they use different statistical values. On the other hand, due to the great variety of attack types, we chose to treat them not as genres belonging to the criterion *dynamic* but as a separated criterion.

Table 2 depicts the intended correlations between Schaefferian criteria (and their attributes expressed in types, classes, genres, or species) and the selected

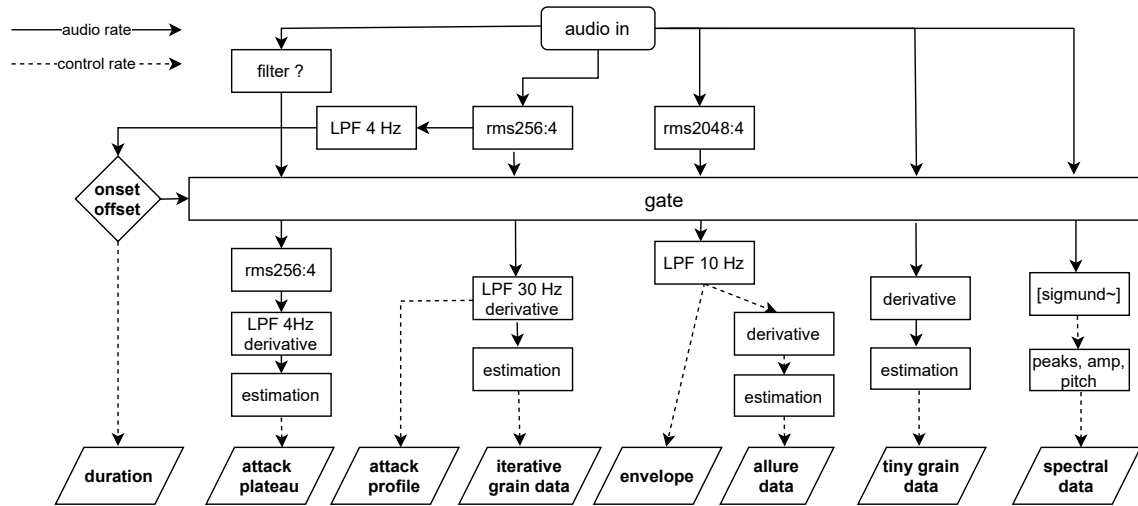


Figure 2: Flowchart showing the main routes and procedures for the estimation of low-level audio descriptors.

Table 2: Intended correspondences between Schaeffer criteria and low-level audio descriptors.

Criterion	Attributes	Audio Descriptors
<i>dynamic</i>	duration: short, formed, long dynamic level: pp to ff dynamic forms: shock, resonance, profiles (5 classes), flat, nil	onset–offset rms curve statistics skewness of spectral centroid spectral region, attack genre
<i>attack profile</i>	genre: abrupt, solid, soft, gentle, stressed, nil	attack profile statistics iterative grain / allure data
<i>mass / harmonic timbre</i>	class: tonic, channeled, nodal group, node region: low, medium, high genre/species: full/hollow/narrow, rich/poor	spectral peaks data / statistics spectral centroid
<i>melodic / mass profiles</i>	density of information: weak, medium, strong type: <i>fluctuation</i> , <i>evolution</i> , <i>modulation</i> ⁶	spectral peaks data / statistics spectral centroid allure data
<i>grain</i>	type: iterative, tiny (friction or resonant) density: rough, matt, smooth	iterative / tiny grain statistics
<i>allure</i>	intensity: weak, medium, strong genre: regular, progressive, irregular, etc.	allure data

low-level descriptors. Note that a few estimated attributes function as descriptors for other categories, like the spectral regions for the dynamic levels or allures for the melodic/mass profiles. It is important to note that any sound object enacts at least three perceptual criteria: mass, dynamic (including the attack), and harmonic timbre. The other criteria may be relevant to characterize specific sounds, and their presence may affect the others. For example, allures may affect the melodic profile, grains may affect the mass and the mass profile, and percussion-resonance types condition the harmonic profile. A comprehensive discussion of the use of the profile concept by Schaeffer (and its implications on our work) is beyond the scope of this paper.

⁶Schaeffer proposes a “typology of variations”. *Fluctuation* is a “variation that is felt only as an imperfection in a desired stability”. *Evolution* stands for a progressive variation. *Modulation* is understood as a variation with “a development in stages — already sketching out a scalar structure.” (TOM[8], p. 453).

6. Examples and Discussion

For every input sound, our program generates real-time curves (or markers) for all descriptors and calculates the scalar values described in section 4. These results are relatively numerous and probably present some degree of redundancy, not yet analyzed. For the sake of clarity, we discuss these results separated by criterion (or sub-criterion), using subsets of sounds and descriptors.

Our first example depicts the quantitative results (Table 3) for the different attack profiles illustrated in Figure 1. These shapes depend on the nature of the sound objects (single stroke, iteration, continuous excitation) and dynamic levels. Schaeffer defines seven genres of attack: abrupt, steep, soft, flat, gentle, sforzando, and nil. The whip sound has a short duration, low values for temporal centroid and spread, a positive skewness, and a high crest. All this data corresponds to an abrupt genre. The slap on a tambourine presents the steep profile. It has a short reso-

nance following the attack. Its temporal centroid is similar to the whip, but with a larger spread and less pronounced values for skewness and crest. A reinforcement of the resonance follows a soft profile, such as with the tuned gong. Here, a longer duration, a slightly positive skewness, and a high value for flatness contribute to the characterization of the genre. We perceive the rattle profile as gentle, given the absence of an initial shock. Its sound production combines iterative and continuous energies, which correlates with a negative skewness, a small crest, and medium flatness.

Two sounds have a clear iterative or granular profile, the guiro rub, and the berimbau jeté. This last has a profile between steep and soft, due to the reinforcement of the resonance by repeated strokes. The guiro profile is *sforzando*, demonstrated by a high temporal centroid, a negative skewness, a high crest. Two sounds bear a nil profile, the bowed vibes, and the thunder shake. This fact is reflected by the high temporal centroid, negative skewness and small crest. There was no estimation of the first plateau during the first 400 ms for the bowed sound. Long sounds will rely less on their attack profile for their qualification. We also believe that the iterative/granular character should be a second-order qualifier for the attack profiles. A similar approach can be made to dynamic classes, analyzing the entire envelope.

Table 4 depicts *allure* data for six sounds, each of them with a different excitation pattern. As already exposed, we have chosen to interpret as allures (and not as new attacks) iterative sustainments with clearly differentiable impulses, as in the cases of the berimbau and sleigh bells. Time intervals with small standard deviation values are related to ordered or regular instances (like the berimbau), while the opposite points to higher irregularity (rain-stick). The amplitudes indicate the depth of variation; for example, the tuned gong resonant allures are much softer than the iterative allures of the sleigh bells. Symmetry indicates the regularity of transitions between peaks and troughs (and vice-versa), and spikiness values point to the suddenness of variation.

The estimation of *grains* for eight sounds is depicted in Table 5. As expected, sounds with iterative sustainment present a large number of these grains. The exception is the bass drum, whose resonant grains, due to the slow rate, also fit into this category. The diverse values for size and duration also helps differentiating between iterative grains. Our tiny grain descriptor is dedicated to resonant and friction grains, and their mixture. A closer inspection of the number of tiny grains and their standard deviation furnish information about their temporal behavior. The large standard deviation, along with a high value of temporal centroid in the bass drum, indicates an increase of background noise at the end of the resonance. The bowed cymbal also displays a considerable standard deviation, but a temporal centroid below 0.5; in this case, the granular characteristic is more present at the beginning. Friction grains tend to have higher bend values than resonant ones, as displayed by the ratchet and cymbal data. On the other hand, the resonant characteristic seems to prevail

in the bowed cymbal and vibraphone, despite the excitation mode.

Seven classes of *mass* are defined by Schaeffer: pure sound, tonic, tonic group, channeled, nodal group, node, white noise. Pure and tonic sounds bear a clear pitch, and tonic groups indicate chord-like sonorities. Node is a filtered noise (or a dense spectral region), while nodal group is a set of nodes. In the middle of this classification we encounter the ambiguous channeled sound, sharing properties for pitched and unpitched classes. The most common classes in the percussive realm are the tonic, channeled, node, and nodal group, although this latter occurs more in combination than in a single object⁷. As stated above, the *harmonic timbre* is a complementary characteristic of the spectral perception, and Schaeffer points to oppositions like hollow/full, rich/poor, and bright/matt. Data related to mass and harmonic timbre is displayed in Tables 6 and 7. The ratio between the unpitched and total frames estimated for the entire sound object points to its tonic character. Low values indicate tonic sounds, like the bass drum, whistle, and the friction of a pandeiro's skin. Higher values of this descriptor, combined with small values of percentiles 50% and 80%, can qualify channeled sounds, like the snare drum (with no snare) and the tuned gong. A high unpitched/total ratio, along with a low value for the energy carried by the 20 first spectral peaks, characterizes nodes, as in the cases of the ratchet and rattle. Spectral centroid and region values are self-explaining.

The intrinsic dissonance values are more ambiguous. Although they may help differentiating between tonic and non-tonic sounds, this is not a univocal association since inharmonic spectral peaks only increment this value if they are close enough in frequency (see [14]). We try to approximate the perceptive attributes *full/hollow/narrow* with Δ peaks, pct50, pct80, and region, and the opposition *rich/poor* with the values estimated for pct80 and 20P/total. For instance, two sounds classified in the medium region may be contrasted through the attributes hollow (snare drum) and narrow (rattle).

High standard deviation values may indicate the presence of audible profiles. The analysis of profiles deserves a dedicated inquiry for the following reasons: (1) the non-univocal use of the concept in the TOM; (2) their variety (dynamic, melodic, mass, harmonic, allures) and interdependency; (3) the choice of the best fitting parameters for each situation — which may require non-realtime dimensionality reduction techniques (PCA, NMF, etc).

7. Final remarks

After presenting a general overview of the setup and some examples, our next steps are directed to a more comprehensive approach of sound profiles, to its implementation in PureData, to the work with musicians in specific instrumental configurations (when the pandemic permits), to the

⁷It is advisable to increase the number of spectral regions for dealing with nodal groups.

Table 3: Attack parameters for eight selected percussive sounds (the same from Figure 1), plus total duration and dynamic level.

sound	FPSlope (dB/ms)	temp. centroid	temp. spread	skewness	kurtosis	crest	flatness	dur (ms)	DL (dB)
whip	0.42	0.24	0.26	1.25	2.87	6.08	0.30	396	-46
tamb.slapp	0.85	0.22	0.74	0.31	0.25	4.07	0.42	461	-33.5
vibes.bow	-	0.71	0.68	-0.35	0.40	2.84	0.70	2177	-26.5
guiro	0.63	0.72	0.39	-0.66	1.36	11.09	0.67	631	-42
rattle	0.13	0.67	0.67	-0.21	0.28	2.18	0.67	1151	-42
berimb.jete	0.81	0.35	0.53	0.33	0.66	4.77	0.60	2019	-55.6
gong.tuned	0.24	0.42	1.13	0.08	0.11	1.96	0.92	9219	-42.6
thunder.shake	0.18	0.53	0.20	-0.19	3.82	2.80	0.86	15541	-33

Table 4: Allure values for six selected percussive sounds.

sound	dur (ms)	number	amp (dB)	Δt (ms)	symmetry	spikiness
berimb.jete	2019	11	5.1 ± 0.8	162 ± 32	1.99 ± 0.94	4.93 ± 3.2
berimb.vib	4687	12	7.3 ± 2.4	349 ± 68	0.99 ± 0.48	1.91 ± 0.79
gong.tuned	9219	6	4.2 ± 1.2	1476 ± 485	1.05 ± 0.56	0.26 ± 11
sleighbells	10295	30	17.1 ± 4.6	333 ± 55	1.47 ± 1	5.34 ± 3.3
thunder.shake	15541	57	7.2 ± 3.8	261 ± 95	1.46 ± 1.2	2.97 ± 2.2
rainstick	17295	27	5.6 ± 3.1	647 ± 448	1.23 ± 1.1	1.77 ± 1.65

Table 5: Grain values for 10 selected percussive sounds.

sound	dur (ms)	iterative grains			tiny grains		
		number	size (dB)	dur (ms)	number	TC	bend (dB)
guiro	631	14	5 ± 3	22.5 ± 13	140 ± 25	0.49	-60.5
ratchet	753	12	15 ± 5	54.5 ± 8	173 ± 18	0.49	-31.0
rattle	1103	2	4.8	73.3	129 ± 18	0.48	-59
pand.rim.frict	1365	3	5.2 ± 1.5	27	218 ± 17	0.5	-29
whistle	1463	30	4.6 ± 2.7	37.5 ± 12	56 ± 11	0.48	-58
bassdrum	3671	96	3.4 ± 1.4	37.4 ± 17	190 ± 122	0.67	-86
cymbal.bow	4565	3	2.7 ± 0.4	34.7 ± 8	65 ± 34	0.44	-76
cymbal	4963	-	-	-	116 ± 48	0.57	-71
tamb.tremolo	7884	59	5.6 ± 1.8	49.5 ± 17	171 ± 11	0.49	-41.5
rainstick	17245	110	6.8 ± 3.3	44.3 ± 16.5	191 ± 32	0.49	-54

Table 6: Mass and harmonic timbre parameters (1) for 10 selected percussive sounds.

sound	dur	pct50	pct80	20P/total (ratio)	unpitched/total (ratio)
tabla.gliss	227	1.3 ± 0.5	6.6 ± 8.5	0.85 ± 0.12	0.24
sdram.nosnare	560	1.1 ± 0.4	2.4 ± 3.7	0.96 ± 0.1	0.32
ratchet	753	19.8 ± 0.6	20 ± 0	0.42 ± 0.1	0.86
rattle	1103	8.4 ± 2.2	19.7 ± 1.3	0.71 ± 0.1	1.0
pand.skin.frict	1915	1 ± 0.2	1.5 ± 2.1	0.98 ± 0	0.07
slidewhistle	641	6.3 ± 7.7	14.4 ± 8.4	0.66 ± 0.5	0.13
chin.opera.gong	1911	2.5 ± 1.9	8.5 ± 5.9	0.9 ± 0.1	0.75
berimb.jete	2019	4.6 ± 5.4	10 ± 6.7	0.82 ± 0.2	0.94
bassdrum	3671	1.1 ± 1	1.2 ± 1.5	0.98 ± 0.1	0.04
gong.tuned	9219	1.3 ± 0.8	2.2 ± 1.6	0.98 ± 0.05	0.42

Table 7: Mass and harmonic timbre parameters (2) for 10 selected percussive sounds.

sound	diss	MPP (mc)	Δ peaks (mc)	SC (mc)	region
tabla.gliss	37 ± 22.5	46.2 ± 10.4	69.9 ± 19.3	54 ± 7.6	1.8 ± 1
sdrum.nosnare	45.3 ± 21.9	61.2 ± 7.7	59.4 ± 15.6	65.5 ± 13.3	3
ratchet	122.5 ± 30.9	97.6 ± 9.5	30.7 ± 4.5	110.2 ± 3.6	6.5 ± 1.3
rattle	138.8 ± 42.6	95.5 ± 1.9	17.5 ± 9.1	101.2 ± 2.7	3
pand.skin.frict	43.2 ± 30.5	48 ± 1.6	70.2 ± 6.7	51.7 ± 6.7	1
slidewhistle	141.1 ± 66.8	89.8 ± 14.8	79.8 ± 20.5	95.3 ± 8.1	4.4 ± 1.8
chin.opera.gong	69.1 ± 15.1	73.2 ± 5.7	40.9 ± 12.6	82.8 ± 6.8	3
berimb.jete	24.8 ± 18.6	65.5 ± 13.9	80.8 ± 13.2	86.5 ± 8.5	3.5 ± 1.3
bassdrum	22.6 ± 23	27.8 ± 3.8	99.3 ± 21.6	30.5 ± 8.7	1 ± 0.3
gong.tuned	24.2 ± 17.8	61 ± 2.9	76.4 ± 31	68.2 ± 7.6	3 ± 0.2

Table 8: Selected sounds.

sound	description
tabla.gliss	single tabla stroke with glissando
whip	single whip attack
tamb.slapp	single tambourine hand slap
sdrum.nosnare	single snare drum stroke, without snare
chin.opera.gong	single chinese opera gong stroke
bassdrum	single bassdrum stroke
cymbal	single cymbal stroke
gong.tuned	single tuned gong stroke
guiro	single directional guiro rub
ratchet	single ratchet swing
rattle	single directional rattle shake
tamb.tremolo	tambourine tremolo
berimb.jete	berimabau jete, multiple strokes
berimb.vib	single berimbau stroke, with vibrato
pand.rim.frict	pandeiro tremolo-like rim friction
sleighbells	multiple sleighbells shakes
thunder.shake	multiple thunder sheet shakes
rainstick	rainstick tip
slidewhistle	slide whistle blow with glissando
pand.skin.frict	single pandeiro skin friction
vibes.bow	single vibraphone key bow
cymbal.bow	single cymbal bow

use of machine learning with larger sets of sounds, and creative applications.

We would like to finish with a quotation from Schaeffer, for us a standing source of caution and stimulus: “It is perhaps disconcerting to see us, after so many warnings, recommending the use of the bathygraph and the Sonagraph to describe a piece of music.” (TOM [8], pp. 556–567)

Aknowledgments

This work is supported by the Brazilian research agency CNPq (National Council for Scientific and Technological Development).

References

- [1] Andrea Valle. Schaeffer Reconsidered: a Typological Space and its Analytical Applications. *Analitica*, 8(1):1–15, 2015.
- [2] Rolf Inge Godøy. Perceiving Sound Objects in the Musique Concrète. *Frontiers in Psychology*, 12, 2021.
- [3] G. Bernardes, M. Davies, and C. Guedes. A Pure Data Spectro-Morphological Analysis Toolkit for Sound-Based Composition. pages 31–38, Aveiro, 2015. Proceedings of the eaw2015.
- [4] Julien Ricard. *Towards Computational Morphological Description of Sound*. Doctorate, Universitat Pompeu Fabra, Barcelona, September 2004.
- [5] Geoffroy Peeters and Emmanuel Deruty. Sound Indexing Using Morphological Description. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):675–687, March 2010.
- [6] Samuel Z. Solomon. *How to Write for Percussion: a Comprehensive Guide to Percussion Composition*. Oxford University Press, New York, 2nd edition, 2016.
- [7] Pierre Schaeffer. *Traité des Objets Musicaux*. Éditions du Seuil, Paris, 1966.
- [8] Pierre Schaeffer. *Treatise on Musical Objects: Essays Across Disciplines*. University of California Press, Oakland, California, 2017.
- [9] Michel Chion. *Guide des Objets Sonores*. Buchet/Chastel, Paris, 1983.
- [10] W. Gieseler, L. Lombardi, and R. Weyer. *Instrumentation in der Musik des 20. Jahrhunderts: Akustik, Instrumente, Zusammenwirken*. Moeck Verlag, Celle, 1985.
- [11] Miller S Puckette, Theodore Apel, and David D Zicarelli. Real-time Audio Analysis Tools for Pd and MSP. In *Proceedings of the International Computer Music Conference*, pages 109–112, San Francisco, 1998.
- [12] G. Peeters, B. L. Giordano, P. Susini, N. Misdariis, and S. McAdams. The Timbre Toolbox: Extracting Audio Descriptors from Musical Signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, November 2011.
- [13] Xavier Serra and Julius Smith. Spectral Modeling Synthesis: A Sound Analysis/Synthesis System Based on a Deterministic Plus Stochastic Decomposition. *Computer Music Journal*, 14(4):12–24, 1990. Publisher: The MIT Press.
- [14] William A Sethares. *Tuning, Timbre, Spectrum, Scale*. Springer, London, 2005.

A Roughness Model Implementation to Analyze Sound Mass Composition: A Case Study in Ligeti's *Continuum* for Harpsichord (1968)

Micael Antunes^{1*}, Guilherme Feulo do Espirito Santo², Jônatas Manzolli^{1†}

¹NICS - Interdisciplinary Nucleus for Sound Studies, University of Campinas
R. da Reitoria, 165 – 13083-872 Campinas, SP

²Computer Science Department, University of São Paulo
Rua do Matão, 1010 – 05508-090 São Paulo - SP

micael.antunes@nics.unicamp.br

Abstract. We present in this paper an implementation of a roughness descriptor based on the model of Pantelis Vassilakis. We use the implemented descriptor to carry out a computer-aided musical analysis of György Ligeti's *Continuum* for harpsichord (1968). Our analysis establishes a parallel between the roughness model, concepts of sound mass composition, microtime perception, and Ligeti's ideas of timbre of movement and permeability. As a result, we display a graphical representation for the roughness computed from the *Continuum*'s recording, propose a formal segmentation for the piece, and analyze its formal development from a musicological point of view.

1. Introduction

In this paper, the implementation of a *roughness* descriptor is presented, aiming at the musical analysis of *sound mass* composition [1]. As a case study, a musical analysis of Ligeti's *Continuum* for harpsichord (1968) was carried out. This approach is justified in this context due to the perceptual characteristics of Ligeti's compositional poetics [2].

The use of audio descriptors in musical analysis is a well-known practice in the context of systematic and empirical musicology [3]. The graphical representation of audio descriptors provides extra insights into the formal aspects of a musical work, offering a complementary perspective with respect to the symbolic information from the score [3, 4]. Different types of audio descriptors have been also applied to investigate emergent and perceptual properties in the electroacoustic and live electronic music repertoire [5, 6, 7, 8].

An important characteristic of the *sound masses* is their *microtime* behavior [9, 10]. *Microtime* refers to a scale of sound perception, which Roads define as “Sound particles on a time scale that extends down to the threshold of auditory perception (measured in thousandths of a second or milliseconds)” [11, p. 4]. *Roughness* is a psychoacoustic model that represents the perception of very fast amplitude fluctuations in sound [12]. Therefore, we

may understand *roughness* as a representative feature of *microtime* perception.

Ligeti's *Continuum* for harpsichord (1968) has been studied through the symbolic analysis of the score according to the *pattern-meccanico* technique [13] and the *Gestalt* model of perception [14]. From an experimental perspective, Douglas et al. used *auditory scene analysis* to study excerpts of *Continuum* and its *sound mass* characteristics [15]. *Microtime* perception in *Continuum* was also investigated using *audio descriptors* [10, 9].

In this study, we propose an implementation for a *roughness* audio descriptor based on Vassilakis' *roughness* model. Also, as a case study, we conduct an analysis of Ligeti's *Continuum* with the implemented descriptor. Considering *Continuum*'s very peculiar *sound mass* characteristics associated with its performance, we investigate how *microtime* features may be represented by the *roughness* descriptor. We describe how such a computer-aided musical analysis may lead to a segmentation of the piece that enhances the understanding of its formal development concerning *sound masses*. Therefore, we extend previous studies based on the symbolic analysis of *Continuum*'s score [13, 14, 15] by investigating singular characteristics of the chosen performance.

In Section 2 we present the central musicological concepts for this work: the idea of *sound mass* composition, Ligeti's concepts of *permeability* and *timbre of movement*, and an overview of his *Continuum* for harpsichord. In Section 3, we present the *roughness* model, its theoretical background, its mathematical formalization, and its proposed implementation. In Section 4 we present the guidelines and methodology for the musical analysis. In Section 5 we display results and in Section 6 we present a conclusion and a discussion about the perspectives for the computer-aided musical analysis of *sound mass music*.

2. Musicological Background

2.1. Sound Mass Composition

Sound mass composition aims to achieve the limits of the perception by exploring a large agglutination of sound in the time and frequency domains [15]. Composers like Karlheinz Stockhausen (1928-2007) have a contribution to the theoretical and aesthetics discussion of *sound*

*Micael Antunes is supported by FAPESP in his doctoral research at Arts Institut of Unicamp/ NICS-UNICAMP, process 2019/09734-3.

†Jônatas Manzolli is supported by CNPq under a Pq fellowship, process 304431/2018-4 and 429620/2018-7.

mass concept [16], and pioneer works with this approach became with composers like Iannis Xenakis (1922-2001), György Ligeti (1923-2006), and Krystof Penderecki (1933-2020). *Sound masses* could be achieved through many compositional techniques, like *sound clusters*, stochastic procedures, granular synthesis, *micropolyphony*, instrumental extended techniques, etc.

According to Douglas et. al. [15], “Sound mass exists when the individual identities of multiple sound events or components are attenuated and subsumed into a perceptual whole, which nevertheless retains an impression of multiplicity”. The authors also point that the *sound mass* is comprised by the complexity of the sound organization, including rhythmic, melodic and spectral parameters of a composition [15]. This concept will be central in our study because the compositional techniques of György Ligeti are strongly related to the concept of *sound mass*. The details of this relationship will be clarified in the next topic.

2.2. György Ligeti’s poetics

György Ligeti (1923 - 2006) is a composer born in Transilvania in a region disputed by Hungary and Romania during the world wars. He is a fruitful composer with a large number of works exploring a diversity of musical techniques and artistic movements. Most of his famous works, like *Atmosphères* (1961) and *Lux Aeterna* (1966), can be associated with the musical concept of *sound mass* composition [1].

Ligeti’s *sound mass* compositions are linked with his interest in natural sciences, mathematics, acoustics, and psychoacoustics [17]. These interests are transposed to his musical thinking under musical concepts like *timbre of movement* and *permeability*.

Timbre of movement refers to the sound fusion, which occurs when the musical texture reaches such a level of complexity that one can not perceive individual events [18, p. 169]. Therefore, the listener’s attention is led to global characteristics and inner movement of the musical texture.

Permeability refers to the absence of pitch perception in the musical texture. According to the composer: “*The loss of sensitivity to intervals is at the source of a state that could be called permeability*” [18, p. 123]. These concepts are operated in his composition by using techniques like *micropolyphony* [19], *pattern-meccanico* technique [13], and *sound clusters* [20]. The *Continuum for harpsichord* (1968) is a representative example of these compositional techniques.

2.3. The *Continuum* for harpsichord

Continuum is a consequence of Ligeti’s interest in machinery, under the concept of *pattern-meccanico* technique [13]. Maybe, the most representative work that illustrates this concept is the *Poème symphonique* for 100 metronomes (1962). In this work, 100 metronomes with different velocities are displayed. The main remarkable

sound result of this piece is the rhythmic behavior, which is a consequence of the emergent synchronicity between metronomes in time [21].

Also, *Continuum* is representative of Ligeti’s interest in aural illusions [2]. According to Ligeti: “*I thought to myself, what about composing a piece that would be a paradoxically continuous sound, something like Atmospheres, but that would have to consist of innumerable thin slices of salami?*” [2]. According to Ligeti, he chose the harpsichord to create this piece because this instrument can play very fast notes, and the plectrum in the strings produces “*quite a loud noise*” [2].

Continuum’s score has only one rhythm with the instruction “*Prestissimo = extremely fast, so that the individual tones can hardly be perceived, but rather merge into a continuum. Play very evenly, without articulation of any sort*”. Therefore, the variation in the piece sound result comes from the variety of melodic profiles and *pattern-meccanico* technique written by Ligeti [13].

The musicological interest of Ligeti’s *Continuum* resides in the manner that the constant patterns of the score lead to the emergence of *sound mass* perceptual characteristics during the performance [9]. According to Knop: “*Peitgen remarks on the compositional strategies that Ligeti applied to create what initially seem to be static textures which, however, develop into more differentiated rhythmical (and melodic) units*” [17]. Knop also highlights that this compositional strategy could be understood under Ligeti’s interest in math and psychoacoustics [17]. So, in this context, we want to focus on the *microtime* perceptual characteristics, through the *roughness*’ descriptor.

3. A model for roughness

3.1. Roughness model

The application of audio descriptors to analyze electronic and instrumental composition was largely explored in the computational musicology context (e.g. [22], [23],[24]). The choice of the descriptor that will be applied in the analysis is strongly related to the musical context of the composition and its musicological interest. We explore the main concepts of the *roughness model* to justify its choice for this analysis.

Previous studies have associated the *roughness* model with *microtime* perception in the context of *Continuum*’s *sound mass* composition [10, 9]. The first one applied the *roughness* model to perform a formal segmentation of the piece [10]. The second one investigates how the *pattern-meccanico* technique, proposed by Clendinning [19] leads to the emergence of *microtime* patterns in perception. By using the *roughness* model to analyze the *sound masses* of Ligeti’s *Continuum*, we aim to describe *microtime* perceptual characteristics [11] of *sound masses* based on inner ear properties.

According to Vassilakis [12], *roughness* is a perceptual feature related to the sense of very fast amplitude

variation of the sound [12, p. 2]. It is partially conditioned by the sound stimulus and partially conditioned by the properties of the basilar membrane. This model is related to the concept of *sensory dissonance* in the context of western music [25, 23], but has also been explored in other non-tonal musical contexts, like Bosnian *Ganga songs* [12, p. 17].

The *roughness* models of Sethares [23] and Vassilakis [12] are based on Plomp and Levelt's dissonance curve [25], which displays a correlation between the sensation of dissonance and Zwicker's critical bandwidth model [26]. According to the results of Plomp and Levelt's experiment [25], the sensation of maximum dissonance for a pair of pure tones occurs when the distance between their respective frequencies is about a quarter of the critical bandwidth size [25]. Based on these results, Sethares [23] developed a model that consists of a parameterized curve for the dissonance values between a pair of pure tones, as we see in Figure 1. The *roughness* curve exhibits a profile with a peak in a quarter of critical bandwidth, which matches in this case with a semitone of the scale. Then, the *roughness* decreases until its minimum, in the final part of the *critical bandwidth*, around the major third of the scale (4 semitones).

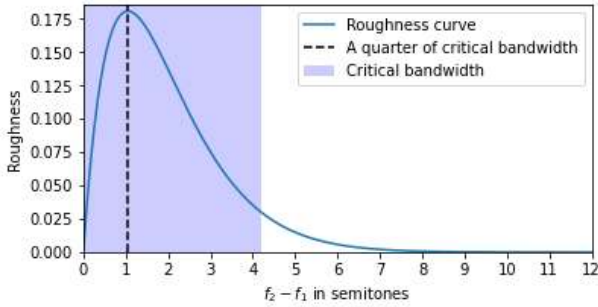


Figure 1: Roughness curve for two pure tones, in function of the twelve-tone equal-tempered scale with fundamental frequency 440Hz.

3.2. Roughness descriptor

The *roughness* value of a time frame is based on an approximation, proposed by Sethares [23] of the Plomp & Levelt's experimental dissonance curve¹ [25], and is defined for two pure sinusoidal tones with frequencies f_1 and f_2 as follows

$$r(f_1, f_2) = e^{\frac{0.84|f_2-f_1|}{0.0207f_1+18.96}} - e^{\frac{1.38|f_2-f_1|}{0.0207f_1+18.96}}. \quad (1)$$

The formulation present in Equation 1 was revised by Vassilakis (Eq. 2) to incorporate the amplitude fluctuation characteristics in the sound stimulus on *roughness* perception, which were underestimated in Sethares' model

$$r(f_1, a_1, f_2, a_2) = (a_i * a_j)^{0.1} \left(\frac{2a_j}{a_i + a_j} \right)^{3.11} r(f_1, f_2), \quad (2)$$

¹For a full revision on *roughness* curves see [12, 27].

where a_1 and a_2 are amplitude values for the f_1 and f_2 components respectively.

For complex sounds, the *roughness* value can be computed using a formulation by Vassilakis [12], which combines all pairwise *roughness* values corresponding to pairs of sinusoidal partials

$$\text{Roughness} = \sum_{i=1}^N \sum_{j=1}^N \frac{r(f_i, a_i, f_j, a_j)}{2} \quad (3)$$

The computational implementation of the *roughness* descriptor is displayed in Algorithm 1. The magnitude spectrum is computed from the input signal x using the Short-Time Fourier Transform (STFT). Based on the spectral content, the frequency values and their respective amplitudes are obtained through peak selection for each time window. The frequency and amplitude values are then used to compute the *roughness* values as described in Equation 3.

Algorithm 1 Roughness descriptor

```

1: function ROUGHNESS(x)
2:    $X \leftarrow |\text{STFT}(x)|$ 
3:    $F, T \leftarrow \text{SIZE}(X)$   $\triangleright X$  is a  $F \times T$  matrix
4:   for  $t \leftarrow 1$  to  $T$  do
5:      $\text{freqs}, \text{amps} \leftarrow \text{PEAK\_PEAK}(\text{frame})$ 
6:      $R[t] \leftarrow 0$ 
7:     for  $f1, a1 \in \text{freqs}, \text{amps}$  do
8:       for  $f2, a2 \in \text{freqs}, \text{amps}$  do
9:          $R[t] \leftarrow R[t] + r(f1, a1, f2, a2)/2$ 
10:  return  $R$ 
```

The source code for the implementation of the *roughness* descriptor and all the graphical analysis was done using Python² and the Jupyter³ environment and are available at this Gitlab repository⁴.

The frequency and amplitudes values were obtained from the magnitude spectrogram using the Librosa [28] library, with a window size of 4096 samples and a hop length of 1024 samples.

4. Methodology

With the following methodology, we aim to apply the *roughness* psychoacoustic model to represent the *microtime* characteristics [11] in the *Continuum* through the analysis of the performance recording shown in Table 1, enlarging the proposal of previous studies [29, 30].

We hypothesize that this descriptor could represent the concept of *timbre of movement* and *permeability*, as well the *microtime* behavior of the *sound masses*. These concepts could be associated with the principles of *minimum masking* and *limited density* [31], which states that

²<https://www.python.org/>

³<https://jupyter.org/>

⁴<https://gitlab.com/Feulo/ligetis-continuum-analysis>

Album	Requiem / Lontano / Continuum
Composer	György Ligeti
Performer	Antoinette Vischer
Year	1968
Country	France

Table 1: Metadata of the chosen performance recording.

the higher the level of spectral information in the auditory nerves, the lower the ability to perceive musical pitches and intervals. These principles of *minimum masking* and *limited density* were already associated with the concept of *sound mass* [1]. Starting from this point of view, we argue that the *roughness* model will provide a one-dimensional representation that will reflect:

- (A) *The density level and agglutination of the spectra in the frequency domain*- The *roughness* model is based on the level of the energy into the *critical bandwidths*, the higher the density and agglutination of the sound spectra, the higher the level of *roughness*.
- (B) *Perceptual features related to microtime amplitude fluctuation*- As already pointed, the *roughness* model presented is based on subjective tests and properties of the inner ear aligned with the subjective perception of sound amplitude fluctuation [12].

Based on the above guidelines, we conducted the musical analysis of the chosen performance recording of the *Continuum* as follows:

1. *Roughness calculation from the audio data*: By applying the algorithm presented in section 3.1, a *roughness* curve of the *Continuum*'s recording will be generated.
2. *Segmentation of the piece*: Based on the *roughness* curve levels, we propose a segmentation of the piece based on the *local minimums* of the curve. It provides an overview of the temporal development of the sound material through the composition.
3. *Musicological considerations*: In this step, we perform the analysis starting from the musicological information about the *Continuum* and the *roughness* descriptor representation.

5. Results

We disclose in this section the results of the analysis, divided into the three steps we present above in section 3.2. The analysis is anchored in the graphical representation of Figure 2, which displays the *roughness* in blue and its respective moving average, for 20 samples, in orange. The formal segmentation of the piece is represented by the dotted vertical black lines, as we will detail below.

1. Roughness calculation

By plotting the *roughness* curve in function of the time of the *Continuum*, we could observe a decreasing profile of the curve through the piece, with the highest value

around 25 seconds, and the lowest values in the final part of the piece. Also, we observe a pattern-like behavior during the piece, with an increasing and decreasing behavior with local maximums and minimums in time.

2. Segmentation of the piece

We propose a segmentation of the *Continuum* using as criteria the local minimums in time, which leads to the following initial time of Table 2 for the segments from A to F, as shown in black dotted vertical lines in the Figure 2.

A	B	C	D	E	F
00:00	01:02	01:41	02:17	02:45	03:25

Table 2: Initial time for each proposed segment of the *Continuum*'s recording.

So, we may observe that the segmentation of the piece highlights the *roughness* patterns inside each of the segments.

3. Human conducted musical analysis

The analysis of the *Continuum* with a *roughness* descriptor allows us to speculate about the perceptual outcome of Ligeti's compositional strategy. First, we may notice an overall strategy: the manipulation of the tension of the *sound mass* through the *roughness*. Second, in terms of the development of the material in time: a kind of pattern in each segment of the piece could be the source of a perceptual coherence in the composition. Third, by observing the descending profile of the *roughness* in the composition, this analysis reinforces the filtering strategy of the musical texture, as already pointed by [9]. Thus, we may conclude that the diversity of the melodic patterns of the *Continuum* does not dazzle a kind of coherence at the perceptual level. This is achieved by an internal coherence in each segment and an overall directionality during the piece.

It is interesting to remember in this context that the *roughness* model is related to the accumulation of energy into the critical bandwidths. So, we may argue that the sound masses allow different levels of pitch perception, as stated by the concept of *permeability*. Thus, strategies of blurring are at a source of the inner movement of the sound masses in the composition.

6. Discussion and Conclusions

In this paper we presented an implementation of a *roughness* descriptor, applying it in the context of Ligeti's *sound mass music*. With that, we privilege a perceptual approach in the musicological study. The *roughness* descriptor is a feature that, combined with other audio descriptors and symbolic information of the score, could provide a broader methodology of musical analysis of the *sound mass* characteristics.

It is important to notice that the segmentation through the *roughness* profile displays a great convergence with the segmentation anchored in a symbolic analysis of the score performed by Clendinning [13]. It means that the

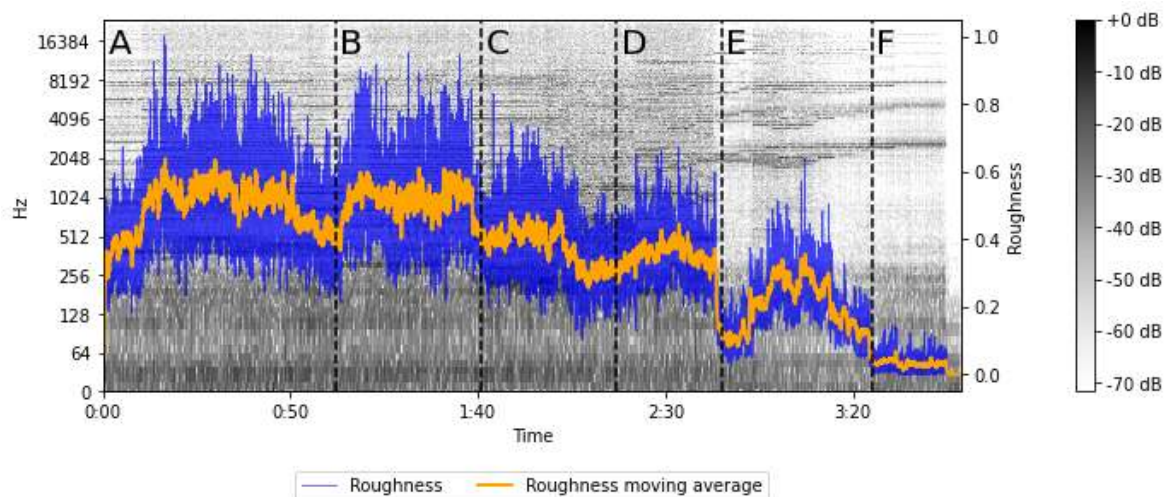


Figure 2: Roughness for the *Continuum*'s recording (blue) and its moving average (orange). The black dotted lines represent the proposed segmentation of the piece. The spectrogram is displayed in the background.

roughness descriptor could provide a meaningful musical representation in the context of *sound mass* composition which explores a great variety of *sound clusters*.

As already demonstrated, the *roughness* profile of the *Continuum* shows an oscillatory pattern through the segments, which is a consequence of the *sound mass* behavior. Therefore, future musical analysis of the *Continuum* could investigate in more detail the relationship between the *pattern-meccanico* technique detected in the symbolic analysis from the score and psychoacoustic patterns obtained through audio descriptors.

Due to the diversity of techniques and instrumental ensembles in the context of *sound mass* music, we need to improve audio signal processing strategies to represent properly the psychoacoustic features. Also, the use of more accurate time-frequency representations could provide better for the *roughness* descriptor computation.

The musical analysis methodology exposed in this paper could be applied in other musicological contexts. Firstly, by enlarging the perceptual knowledge of a musical performance from the audio data, the *roughness descriptor* fills a gap in the context of psychoacoustic descriptors that could be particularly interesting to the musicological field. Second, the *roughness* provides relevant information about the energy concentration of the frequencies in the musical texture that could be used to enlarge the understanding of invariant properties of the *sound masses*.

Future works may perform a comparative analysis between the audio features of the *Continuum*'s performance with a symbolic analysis of the score, to verify how the melodic patterns lead to these psychoacoustic characteristics. Also, experimental works with expert and non-expert listeners could provide more information about the accuracy of the *Roughness* feature in the segmentation of the piece.

References

- [1] Jason Noble and Stephen McAdams. Sound mass, auditory perception, and 'post-tone' music. *Journal of New Music Research*, 49(3):231–251, May 2020.
- [2] György Ligeti and Péter Várnai. *Ligeti in conversation*. Da Capo Press, 1983.
- [3] Stephen McAdams, Philippe Depalle, and Eric Clarke. Analyzing musical sound. In *Empirical musicology: Aims, methods, prospects*, pages 157–196. Oxford University Press, Oxford, 2004.
- [4] Pierre Couprie. Graphical representation: an analytical and publication tool for electroacoustic music. *Organised Sound*, 9(1):109–113, 2004.
- [5] Danilo Rossetti, William Teixeira, and Jonatas Manzoli. Emergent Timbre and Extended Techniques in Live-Electronic Music: An Analysis of Desdobramentos do Contínuo Performed by Audio Descriptors. *Revista Música Hodie*, 18(1), 2018.
- [6] Danilo Rossetti, Micael Antunes, and Jônatas Manzolli. Compositional Procedures in Electronic Music and the Emergence of Time Continuum. *Organised Sound (pre-print)*, 2020.
- [7] Mikhail Malt and Emmanuel Jourdan. Le "BSTD"—Une représentation graphique de la brillance et de l'écart type spectral, comme possible représentation de l'évolution du timbre sonore, 2009.
- [8] Ivan Eiji Simurra and Rodrigo Borges. Combining Automatic Segmentation and Symbolic Analysis based on Timbre Features – A Case Study from Ligeti's Atmospheres. In *Proceedings of the 11th International Conference of Students of Systematic Musicology*, page 4, 2019.
- [9] Micael Antunes, Danilo Rossetti, and Jônatas Manzolli. Emerging structures within micro-time of Ligeti's Continuum (pre-print). In *Proceedings of the 2021 International Computer Music Conference*, Santiago, Chile, 2021.
- [10] Micael Antunes, Guilherme Feulo, and Jônatas Manzolli. A perceptual approach to Ligeti's Continuum with a Roughness descriptor. In *Livro de Resumos do II EINEM Encontro Internacional de Investigação de Estudantes em Música e Musicologia*, pages 18–19, Évora, 2020.
- [11] Curtis Roads. *Microsound*. MIT Press, Cambridge, Mass, 2001.

- [12] Pantelis Nestor Vassilakis. *Perceptual and physical properties of amplitude fluctuation and their musical significance*. PhD Thesis, University of California, Los Angeles, Califórnia, 2001.
- [13] Jane Piper Clendinning. The Pattern-Meccanico Compositions of György Ligeti. *Perspectives of New Music*, 31(1):192, 1993.
- [14] Emílios Cambouropoulos and Costas Tsougras. Auditory Streams in Ligeti’s Continuum: A Theoretical and Perceptual Approach. *Journal of interdisciplinary music studies*, 3(1-2):119–137, 2009.
- [15] Chelsea Douglas, Jason Noble, and Stephen McAdams. Auditory Scene Analysis and the Perception of Sound Mass in Ligeti’s Continuum. *Music Perception*, 33:287–305, 2016.
- [16] Karlheinz Stockhausen and Robin Maconie. *Stockhausen on Music Lectures and Interviews*. Marion Boyars Publishers Ltd, London, 1989.
- [17] Frederik Knop. Making it home? The natural sciences as a site of belonging in György Ligeti’s music. In *György Ligeti’s Cultural Identities*. Routledge, New York, 2018.
- [18] György Ligeti. *Neuf essais sur la musique*. Éditions Contrechamps, Genève – Suisse, 2010.
- [19] Jane Piper Clendinning. *Contrapuntal techniques in the music of György Ligeti*. PhD thesis, Yale University, 1990.
- [20] György Ligeti and Bernard Jonathan W. States, Events, Transformations. *Perspectives of New Music*, 31(1):164, 1993.
- [21] Eric Drott. Ligeti in Fluxus. *Journal of Musicology*, 21(2):201–240, April 2004.
- [22] Pierre Couprie. EAnalysis: aide à l’analyse de la musique électroacoustique. 2012.
- [23] William Sethares. *Tuning, timbre, spectrum, scale*. Springer, London, 2nd ed edition, 2005.
- [24] Micael Antunes, Danilo Rossetti, and Jônatas Manzolli. A computer-based framework to analyze continuous and discontinuous textural works using psychoacoustic audio descriptors. In *Proceedings of the 17th Brazilian Symposium on Computer Music* Federal, pages 4–11, São João del Rey, 2019.
- [25] Reinier Plomp and Willem Johannes Maria Levelt. Tonal consonance and critical bandwidth. *The journal of the Acoustical Society of America*, 38(4):548–560, 1965. Publisher: Acoustical Society of America.
- [26] Eberhard Zwicker, Gordon Flottorp, and Stanley Smith Stevens. Critical Band Width in Loudness Summation. *The Journal of the Acoustical Society of America*, 29(5):548–557, May 1957.
- [27] Micael Antunes. *Redução da dissonância sensorial em uma escala temperada utilizando timbres inarmônicos: uma abordagem experimental e aplicações artísticas*. Dissertação de Mestrado, Universidade de São Paulo, São Paulo, 2018.
- [28] Brian McFee, Colin Raffel, Dawen Liang, Daniel Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. *librosa: Audio and Music Signal Analysis in Python*. pages 18–24, Austin, Texas, 2015.
- [29] Micael Antunes and Jônatas Manzolli. A psychoacoustical approach to Ligeti’s concept of permeability. In *Livro de Resumos do II EINEM Encontro Internacional de Investigação de Estudantes em Música e Musicologia*, pages 20–21, Évora, 2020.
- [30] Micael Antunes, Danilo Rossetti, and Jônatas Manzolli. A computer-aided analysis of the Orchestration in Iannis Xenakis’ Aroua (pre-print). In *Proceedings of the 10th European Music Analysis Conference*, Moscow, 2021.
- [31] David Huron. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception: An Interdisciplinary Journal*, 19(1):1–64, 2001.

Concatenative Sound Synthesis as a Technomorphic Model in Computer-Aided Composition

Júlio Guatimosim¹, José Henrique Padovani¹, Carlos Guatimosim¹

¹LaPIS / CrIS – School of Music/Federal University of Minas Gerais
Av. Antônio Carlos, 6627 – 31270-901 Belo Horizonte, MG

juliogn@ufmg.br, jhp@ufmg.br, cguatimosim@alumni.usp.br

Abstract. The text presents a process aimed at computer-aided composition for percussion instruments based on Concatenative Sound Synthesis (CSS). After the introduction, we address the concept of “technomorphism” and the influence of electroacoustic techniques in instrumental composition. The third section covers processes of instrumental sound synthesis and its development in the context of Computer-Aided Composition (CAC) and Computer-Aided Music Orchestration (CAMO). Then, we describe the general principles of Concatenative Sound Synthesis. The fifth section covers our adaptation of CSS as a technomorphic model for Computer-Aided Composition/Orchestration, employing a corpus of percussion sounds/instruments. In the final section, we discuss future developments and the main characteristics of our implementation and strategy.

1. Introduction

Since the remarkable technological transformation triggered by electroacoustic instruments and techniques, new instrumental composition strategies emerged [1]. At the same time, the development of digital computers led to the development of algorithmic compositional practices, which culminated in research on computer music and the development of compositional practices associated with algorithmic and computational resources.

While the first process led to what is now called *technomorphisms* in musical composition, the second process led to the development of Computer-Aided Composition (CAC) and, later, to the establishment of areas such as Music Information Retrieval (MIR) and Auditory Scene Analysis (ASA): fields that require the automatic detection of features from audio files and streams.

In the present work, we address a general review of CAC and Computer-Aided Music Orchestration (CAMO) processes and the concept of *technomorphism* itself. Then, we describe the implementation of a “technomorphic” approach to music composition, aimed at percussion instruments, based on the technique of digital synthesis called Concatenative Sound Synthesis (CSS). After presenting a general review of the concept of *technomorphism* and Instrumental Sound Synthesis processes in the context of computer-aided orchestration/composition, we outline the principles of CSS to, then, present our technomorphic implementation of this technique. In the end, we discuss the potential of the implemented process, its future consequences, and the general characteristics of the approach used in this work.

2. Technomorphism in music composition

The influence of electroacoustic techniques on instrumental music at the beginning of the second half of the 20th century is notable in the work of composers whose creative work encompassed these two practices. In this context, it is not rare the application of technical means as models or metaphors for procedures used in instrumental compositions that do not use directly those tools. While having broader applications in fields of psychology and philosophy of sciences [2], the term *technomorphism* subsumes this creative approach regarding technological tools in composition. Briefly, it refers to the “the metaphoric use of a technological process applied in a different environment from that in which it was conceived”[3]: in the context of post-war instrumental music, it denotes, particularly, the use of compositional concepts and procedures built upon analogies with equipment, tools, and techniques introduced by electroacoustic technologies[1].

In a broader sense, in line with Gilbert Simondon’s theoretical contribution [4], *technomorphism* is a process of interpretation, appropriation, and resignification of technologies embedded in modern technical objects. Just as the inventor sets down ideas and human gestures in the mechanisms and modes of operation of technical objects, *technomorphism* involves a creative reinvention of these same techniques. It relies, at first, on deciphering the mechanisms, gestures, and ideas set down in the gears and elements of the technical objects, on understanding their functionalities and characteristics. In a second moment, it involves the transposition of these features to the universe of artistic creation, where they may become compositional models, procedures, or materials¹.

With the remarkable development of computational methods for signal analysis and synthesis in recent

¹The relationship between human imagination and the process of operation of technical objects is particularly evidenced by Simondon, in [4]: “To invent is to make ones thought function as a machine might function, neither according to causality, which is too fragmentary, nor according to finality, which is too unitary, but according to the dynamism of lived functioning, grasped because it is produced, accompanied in its genesis. The machine is a being that functions. Its mechanisms concretize a coherent dynamism that once existed in thought, which were that thought. During invention, the dynamism of thought converted itself into functioning forms. Inversely, the machine, in functioning, is subject to or produces a certain number of variations around the fundamental rhythms of its functioning, arising from its definite forms. These variations are what are significant, and they are significant with respect to the archetype of functioning, which is that of thought in the process of invention. One has to have invented or reinvented the machine if the machines variations of functioning are to become information.” [p. 151]

years, it is possible to imagine compositional processes using algorithms that have as technomorphic models new approaches to computational sound synthesis/processing. In the present work, we use DSP analysis resources to implement technomorphic instrumental compositional procedures based on CSS techniques (synthesis/sampling methods that rely on computational resources for analyzing and manipulating audio signals).

3. Instrumental synthesis and computer-aided orchestration/composition

While the basic operations of serial music may be compared to the operational mechanism of computers and Turing machines, the term *technomorphism*, although applicable to the appropriation of any technologies, is usually associated with electronic means of sound synthesis. This association is presumably due to the remarkable impact that the techniques of electroacoustic music had on composers since the post-war period. The influence of these techniques is manifest in orchestral literature since at least the late 1950s, with the decisive influence of electronic processes in pieces such as *Apparitions* and *Atmosphères*, by György Ligeti. From the 1970s onwards, composers of the so-called *spectralisme*, such as Gérard Grisey and Tristan Murail, have systematically explored technomorphic processes based on techniques such as additive synthesis, feedback processes, ring modulation, and FM synthesis. The influence of studio techniques and apparatus on compositional writing of this period is described in detail in [1].

With the rapid expansion and dissemination of computational technologies several composers turned to algorithmic approaches of computer-assisted orchestration and instrumental composition. In general, these efforts have been focused on what composers such as Gérard Grisey [5, 6] and Clarence Barlow [7, 8] came to call, based on diverse aesthetic trajectories and creative elaborations, *instrumental synthesis*.

In general, such approaches consist in treating instrumental writing in resemblance with additive synthesis techniques: it consists, basically, of transcribing into the notes of the instrumental ensemble, the parameters of sinusoidal components whose specific values of frequency and amplitude can be either established through abstract processes/calculations – the case, in general, of instrumental synthesis in the works of spectralism, in the 1970s –, or from the previous analysis of audio signals using different techniques (in particular, the Fast Fourier Transform).

With the growing use of computers and digital technologies from the 1980s onwards, such researches are gradually enhanced by CAC practices. These new approaches to composition and orchestration gradually shift from very personal approaches (such as Gottfried M. Koenig's *Project 1* and *Project 2* [9]) to the employment of digital computers to more broad-use languages, libraries, and CAC environments, intended to help composers to explore algorithmic tools in their creative projects.

In the 2000s, these computational tools for music composition are already mature, being applied in creative contexts not anymore confined to the institutional borders of big European Research and Cultural Centers, like IRCAM. In this context, it is also possible to observe the remarkable development of new areas, like Music Information Retrieval, with a strong emphasis on new DSP algorithms that enable the inference of sonic characteristics. From this, there is the emergence of researches that explore the use of DSP-based processes and corpus of audio samples to open new venues for CAC and CAMO. In [10], for instance, large sets of audio files are analyzed through FFT to extract spectral peak, stored in Common Lisp lists. Target sounds are processed similarly, being subsequently compared to the database. The algorithmic process focuses on finding instrumental timbres that match, with the lowest deviation, the spectral peaks (frequency regions and amplitude) of the given sound file, allowing for the instrumental mimesis of the target audio.

Subsequent works use other DSP analysis methods to extract relevant characteristics of the instrumental corpus and the target sounds [11]. Similarly, they employ strategies to filter intermediate candidates, reducing the space search, enabling optimizations regarding computation time [12]. Finally, these works explore different algorithms and programming strategies to both perform the heuristic search and, also, to rank possible solutions. Among such approaches, it is remarkable the application of bioinspired algorithmic methods such as evolutionary algorithms, artificial immune systems, and neural networks, for instance [13, 14, 15, 16].

While the main objective of such works is, in general lines, to create tools and strategies for target-based imitative orchestration, other researches have been focusing on the creative exploration of audio descriptors in orchestral composition. In such contexts, feature-related data can be applied, for instance, as criteria to qualify, select, or manipulate instrumental sounds and combinations by taking into account characteristics that are related to high-level attributes (such as “brightness” and “roughness” for instance) [17].

4. Concatenative Sound Synthesis

While such approaches are fruitful, enabling the exploration of new creative possibilities for CAC and CAMO, we propose a third approach. Our study aims to develop algorithmic tools/strategies for computer-aided music notation based on CSS and the underlying process of juxtaposing or overlapping sound fragments selected according to their sonic features. This paradigm, already explored consciously without the direct use of computers by composers such as Helmut Lachenmann [18], consists of a different approach of exploring descriptor-based techniques to generate combinations of percussion sounds. Having established the notational resources to indicate such sounds, it becomes possible to use this process to generate percussion score excerpts.

CSS is a computational data-driven synthesis

technique with two inputs: a target sound and a collection of audios from a database. The output is a synthesized sound generated by the concatenation of audio grains², which are selected from a corpus according to the similarity between their features and those of the segments of given target sounds. CSS works by segmenting the input sounds into small audio “units” and analyzing them through audio descriptors and DSP algorithms. A unit selection algorithm measures the distance, in the descriptors’ space, between the feature vectors obtained from the target sound and the sample corpus units [20]. The latter corresponding to the vectors with the smallest distances are sequenced and concatenated to generate the synthesized audio.

Most implementations and applications of CSS use these techniques in processes like musical mosaicing/micromontage [21, 19], sound texture synthesis [22], and real-time signal processing [23, 24]. However, there are still few researches, like [20], that apply CSS concepts and processes in contexts of generative, algorithmic, or computer-aided composition – particularly those focused on instrumental music.

5. Technomorphic Implementation of Concatenative Sound Synthesis

The application of CSS as a technomorphic model for instrumental composition is based on the correlation between relevant features of audio frames of a target sound and those of samples available in a database/corpus. Instead of using the corpus samples to synthesize sounds as in usual CSS processes, the technomorphic adaptation of this synthesis technique uses the respective symbolic information of corpus sounds to inform or generate musical notation. The resulting notation prescribes instrumental indications and mixtures whose resulting sounds are expected to have a degree of similarity to the spectral and temporal structure of the target audio.

In our study, we structured the architecture and the main stages of the technomorphic adaptation of CSS. Our implementation of the process uses the Python programming language due to the diversity of available libraries, packages, and modules. Among the libraries that we are already exploring for our purpose or that may be helpful to further developments of our research are those dedicated to audio analysis (librosa, pyAudioAnalysis, Essentia), musical score notation/manipulation (abjad, music21), and DSP/synthesis (audiolazy, pyo, supriya). Beyond those, popular packages dedicated to machine-learning and linear algebra processes available in Python are promising for descriptor-based processes in CAC and CAMO contexts.

Currently, our implementation comprises four steps:

1. segmentation of both the target sound and of the

²A more detailed explanation of the mechanism of CSS and the differences between it and granular synthesis can be found at [19].

database’s audio samples into *attack* and *decay* segments;

2. analysis of the segments by a set of audio descriptors;
3. selection/ordering of corpus samples segments according to the degree of similarity between them and those of the target sound;
4. music notation/transcription according to the selected corpus samples.

The following subsections describe in more detail this preliminary approach to CSS as a technomorphic model. In our tests, the target audio is an electroacoustic soundscape consisting of different percussive/impulsive events. The samples corpus is composed of simple percussion instrument samples (conga, bongo, tom-tom, snare, cowbell, shakers and cymbal).

5.1. Segmentation

Initially, every audio is pre-processed by a method of Harmonic Percussive Source Separation³ [25, 26], which can accentuate transient and attenuate harmonic sounds. Further, a segmentation algorithm computes the root mean square (RMS) envelope of each audio file and applies an onset detection function⁴ and a peak-tracking function⁵. The purpose is to identify individual *sound objects*. Since we have so far worked with relatively simple percussive/impulsive sounds, it is considered that each *sound object* comprises two main parts: the *attack* and the *decay* [27].

Supposing that there are meaningful spectral variations between these two parts that could enrich the diversity of the data, the algorithm splits each sound object into an *attack* and a *decay* fragment according to the location of the onset, peak, and offset frames. The *attack* segment corresponds to the frames between the onset and the peak of the RMS envelope. The *decay* segment spans from the peak frame until an offset frame with an RMS value inferior to an arbitrary threshold⁶ (Figure 1). However, since the target sound can comprehend sustained events or sequences of overlapping gestures, the RMS envelope of the segmented sound objects may not always present a value below the determined offset threshold. In these cases, the *decay* section will span until the frame right before the onset of the following event (Figure 2).

5.2. Analysis

After splitting every *sound object* into an *attack* and a *decay* segment, an algorithm built upon a determined set of audio descriptors analyzes frames of the samples to extract feature-related data as vectors. In this specific test, the audio analysis produced for each frame a vector containing 13 MFCC and 1 RMS coefficient. If an audio fragment

³For this, we used the function `effects.hpss()`, from Librosa.

⁴Namely, `onset.onset_backtrack()`, from Librosa.

⁵Namely, `util.peak_pick()`, from Librosa.

⁶The offset is computed considering RMS envelopes of the original audios since the pre-processed ones would have their reverberation considerably attenuated.

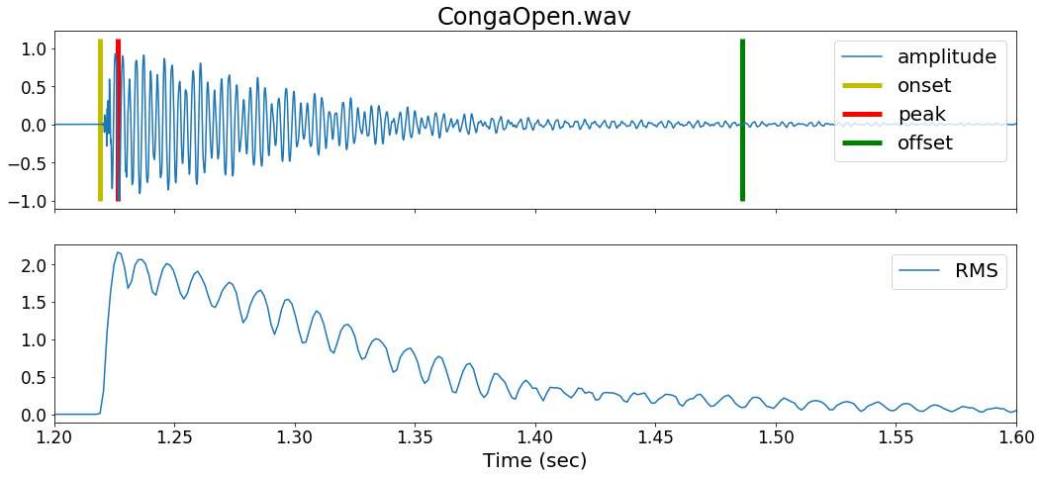


Figure 1: Segmentation of a one-hit conga sample.

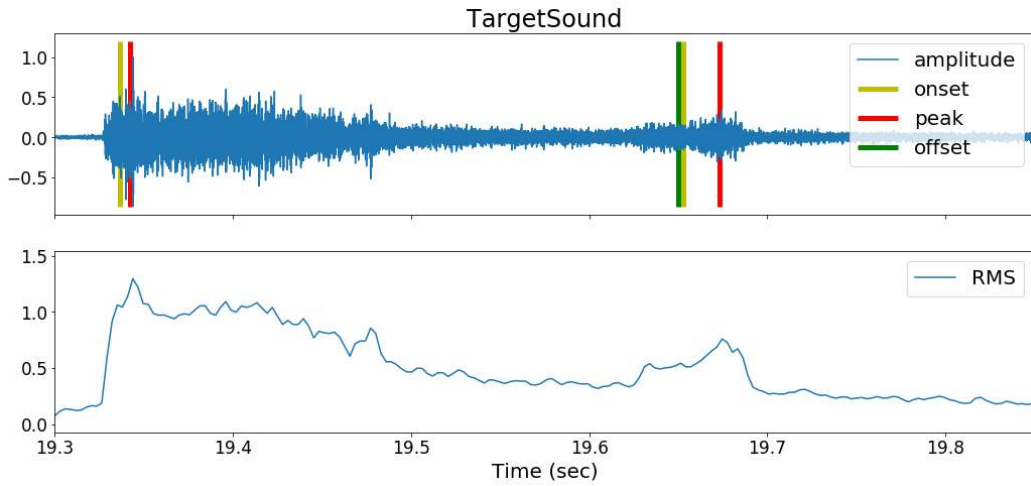


Figure 2: Segmentation of an extract of the target sound.

comprises two or more frames, a function averages corresponding coefficients into one vector.

Formally, let Ω_c and Ω_T be the abstract space of the corpus samples and target sound objects after their segmentation into *attack* and *decay* segments. Then, an element β in those spaces would be understood as a pair (β_1, i) where β_1 represents the sound object and $i \in \{0, 1\}$, where 0 represents the *attack* component of the sound and 1 the *decay*.

We first transform these spaces into a feature space. The vectorization is done by summarizing audio information using audio descriptors, where each of these descriptors will be treated as a dimension. In other words⁷,

$$f : \Omega_T \longrightarrow [0, 1]^d \times \{0, 1\}, \quad (1)$$

$$y_i \triangleq f(\beta) = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_d, i)$$

where $i \in \{0, 1\}$, d is the number of descriptors, and the entries $\bar{y}_j$ are the average of the j -th descriptor of the vectors extracted from each frame of β .

⁷Since we are using the same descriptors for Ω_c , the transform for this space follows analogously.

Notice that the vectorial space is normalized. Since we will use distances to approximate feature vectors, we need to rescale the space to avoid bias towards certain dimensions.

5.3. Selection

Our objective is to approximate elements of Ω_T using Ω_c . In order to convey a notion of distance between the spaces, we chose an Euclidean based approach.

Let $y_i \in f(\Omega_T)$ be a sound object which we wish to approximate using elements of $f(\Omega_c)$. Then such minimizer could be defined as

$$x_i^* \triangleq \arg \min_{x_j \in f(\Omega_c), j=i} D(y_i, x_j) \quad (2)$$

where D is some distance function.

Since Ω_c is a finite space, we may simply use an algorithm to exhaustively search Ω_c , calculate the distances between the elements and y_i and choose the minimizer. Notice that the optimization is restricted to sounds of same kind (attack or decay).

The percussion fragments related to the vectors with the smallest distances are then selected and sequenced according to the respective target sound segments (Table 1). The information related to the origin and kind of the segments is embedded in their names, which follow the given structure:

`<sloc>_<fname>-<kind>_<sdur>`

where `<sloc>` is the fragment location, in samples, in the audio file from which it was extracted, `<fname>` is the name of the audio file, `<kind>` labels the fragment as attack (ATT) or decay (DEC), and `<sdur>` is the segment duration, in samples.

Target Sound Fragment	Distance	Selected Percussion Fragment
22660.TS-ATT_3914	0.169	23690.Cowbell1-ATT_2266
26574.TS-DEC_12772	0.133	44496.SnareBright-DEC_8240
39552.TS-ATT_1442	0.792	23896.SnareBright-ATT_412
40994.TS-DEC_7004	0.076	24576.CymbalStrike2-DEC_219136
...	...	...

Table 1: Attack and decay percussion segments sequenced according to target sound fragments. Distances are normalized between 0 and 1.

To check for acoustic similarities between the target sound and the instrumental sequence, the latter is concatenated into an audio file, as would occur in a traditional process of CSS. This procedure allows the composer to grasp, by hearing, what types of sonorities were selected according to the parameters passed to the segmentation and analysis algorithms. Thus, the composer may experiment with different parameters, such as distinct sets of audio descriptors, and then listen to the acoustic differences between the segment sequences produced in each run before choosing one or more of them to apply in a compositional process. This approach seems interesting to us as it allows flexible applications of the computational tool in compositional contexts. Such an approach enables both generative/algorithmic applications and more dialogical explorations of the computational processes, standing precisely between the extremes of “manual composition” and the “composing machine” that, as Laske points out, situate CAC [9].

5.4. Notation

After these stages, the final step consists of determining the appropriate symbols and resources to notate the percussion sounds in a musical score. This is done using Python and LilyPond data structures and functions to create the music score structure, notational strategies (including rhythm quantization and instrumental technique indications, for instance), and the association between corpus sounds and graphic symbols.

This final step uses the flexibility of Abjad [28, 29], a Python package that enables the algorithmic manipulation of symbolic music structures that are subsequently rendered using LilyPond [30]. While requiring the further formalization of compositional decisions, this approach allows for a highly customizable and flexible algorithmic notation. Figure 3 demonstrates an example of a raw percussion score generated through this process.

6. Conclusion

We have documented a preliminary approach to CSS as a technomorphic model for music composition. Due to the notational purpose, instead of segmenting input audios into units of fractions of a second, as commonly occurs in CSS, a segmentation algorithm splits each audio into *attack* and *decay* fragments. The segments of a target sound are analyzed and compared to those of percussion samples (corpus), allowing for the sequential selection of sound combinations and the respective data related to their duration, instrumentation, and temporal onset/offset. These parameters enable the creative use of the generated data in compositional contexts, enabling CAC approaches to the technomorphic adaptation of CSS.

By using open-source tools and high-level programming and analysis resources, the purpose of our study is to explore descriptor-based CAC/CAMO processes that allow composers and researchers to manipulate particularities of the diverse algorithmic tools involved. Thus, while our approach does not intend to offer a ready-to-use tool for target-based orchestration, it proposes an exploratory integration of different computer tools in the context of feature-based computer-aided composition. Besides a more conscious manipulation of the computational tools applied in CAC/CAMO contexts, this strategy seems to have the potential to integrate research and artistic practices and to enable creative efforts in using other computational tools, such as machine-learning packages/libraries for example.

Regarding the presented implementation, we plan to keep a creative and flexible approach to this model. Further developments comprise other strategies of segmentation for the processing of instrumental samples other than percussion, the structuration of larger sample corpus (using specialized database formats), different approaches of instrument superposition (allowing for the creation of instrumental textures to create “sound types” and orchestrations), and the exploration of machine-learning and other heuristic processes.

References

- [1] Tatiana Catanzaro. *Transformações na linguagem musical contemporânea instrumental e vocal sob a influência da música eletroacústica entre as décadas de 1950-70*. 7 Letras, Rio de Janeiro, RJ, 2018.
- [2] Bryan Holmes. *Tecnomorfismo em música: uma visão teórica e prática*. PhD thesis, Unirio, Rio de Janeiro, 2019.

- [3] Peter Niklas Wilson. Vers une 'écologie des sons': Partiels de Gérard Grisey et l'esthétique du groupe de l'Itinéraire. *Entretiens*, 8:56–81, 1989.
- [4] Gilbert Simondon. *On the mode of existence of technical objects*. Univocal Pub, Minneapolis, MN, 2017.
- [5] Gérard Grisey. À propos de la synthèse instrumentale. In Guy Lelong, editor, *Écrits ou L'invention de la musique spectrale*. MF, Paris, 2008.
- [6] Gérard Grisey. Structuration des timbres dans la musique instrumentale. In Guy Lelong, editor, *Écrits ou L'invention de la musique spectrale*, pages 89–120. MF, Paris, 2008.
- [7] C. Barlow. On the spectral analysis of speech for subsequent resynthesis by acoustic instruments. *undefined*, 1998.
- [8] Clarence Barlow. Music Derived from Other Sources. *Journal of Science and Technology of the Arts*, 13(1):21–44, April 2021. Number: 1.
- [9] Otto Laske. Composition Theory in Koenig's Project One and Project Two. *Computer Music Journal*, 5(4):54–65, 1981. Publisher: The MIT Press.
- [10] David Psenicka. SPORCH: An Algorithm for Orchestration Based on Spectral Analyses of Recorded Sounds. In *Proceedings of the 29th International Computer Music Conference*, pages 1–4, Singapore, 2003. Michigan Publishing.
- [11] Grégoire Carpentier, Damien Tardieu, Gérard Assayag, Xavier Rodet, and Emmanuel Saint-James. Imitative and Generative Orchestration Using Pre-analyzed Sound Databases. In *Proceedings of the Sound and Music Computing Conference*, pages 1–8, Marseille, 2006.
- [12] Grégoire Carpentier and Jean Bresson. Interacting with Symbol, Sound, and Feature Spaces in Orchidée, a Computer-Aided Orchestration Environment. *Computer Music Journal*, 34(1):10–27, March 2010.
- [13] José Abreu, Marcelo Caetano, and Rui Penha. Computer-Aided Musical Orchestration Using an Artificial Immune System. In *Proceedings of the 5th International Conference on Evolutionary and Biologically Inspired Music, Sound, Art and Design - Volume 9596*, pages 1–16, Berlin, Heidelberg, March 2016. Springer-Verlag.
- [14] Jon Gillick, Carmine-Emanuele Cella, and David Bamman. Estimating Unobserved Audio Features for Target-Based Orchestration. In *ISMIR*, 2019.
- [15] Carmine Emanuele Cella, Daniele Ghisi, Vincent Lostanlen, Fabien Lévy, Joshua Fineberg, and Yan Maresz. OrchideaSOL: a dataset of extended instrumental techniques for computer-aided orchestration. *arXiv:2007.00763 [cs, eess]*, July 2020. arXiv: 2007.00763.
- [16] Marcelo Caetano and Carmine E. Cella. Imitative Computer-Aided Musical Orchestration with Biologically Inspired Algorithms. In Eduardo Reck Miranda, editor, *Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity*, pages 585–615. Springer International Publishing, Cham, 2021.
- [17] Ivan Eiji Yamauchi Simurra. *Contribuição ao problema da orquestração assistida por computador com suporte de descritores de áudio*. Doctor of Music Thesis, Universidade Estadual de Campinas, Campinas, 2016.
- [18] Helmut Lachenmann. Klangtypen der Neuen Musik. In *Musik als existentielle Erfahrung: Schriften 1966-1995*, pages 1–20. Breitkopf & Härtel, Wiesbaden, 1996.
- [19] Bob L. Sturm. Adaptive Concatenative Sound Synthesis and Its Application to Micromontage Composition. *Computer Music Journal*, 30(4):46–66, December 2006.
- [20] Gilberto Bernardes de Almeida. *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*. Doctor of Philosophy, University of Porto, Porto, 2014.
- [21] Aymeric Zils and François Pachet. Musical Mosaicing. *Proceedings of the COST G-6 Conference on Digital Audio Effects*, pages 1–6, 2001.
- [22] Diemo Schwarz and Axel Roebel. Concatenative sound texture synthesis methods and evaluation. In *Proceedings of the 19th International Conference on Digital Audio Effects (DAFx-16)*, Brno, Czech Republic, 2016.
- [23] William Brent. *A timbre analysis and classification toolkit for pure data*. Ann Arbor, MI: MPublishing, University of Michigan Library, 2010.
- [24] Diemo Schwarz, Gregory Beller, Bruno Verbrugghe, and Sam Britton. Real-Time Corpus-Based Concatenative Synthesis With CATART. In *Proceedings of the 9th Int. Conference on Digital Audio Effects*, pages 1–7, Montreal, Canada, 2006.
- [25] Derry Fitzgerald. Harmonic/Percussive Separation Using Median Filtering. In *Proceedings of the 13th International Conference on Digital Audio Effects*, pages 1–4, Graz, Austria, 2010.
- [26] Jonathan Driedger, Meinard Müller, and Sascha Disch. Extending Harmonic-Percussive Separation of Audio Signals. In *Proceedings of the 15th International Conference on Music Information Retrieval (ISMIR)*, pages 611–616, Taipei, Taiwan, 2014.
- [27] 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.
- [28] Trevor Baca, Josiah Wolf Oberholtzer, J. Trevino, and V. Adán. Abjad: An open-source software system for formalized score control. In *Proceedings of The First International Conference on Technologies for Music Notation and Representation*, 2015.
- [29] Gregory Evans. *An Introduction to Modeling Composition Through Abjad's Model of Music Notation*. Master's thesis, University of Miami, Miami, 2019.
- [30] Han-Wen Nienhuys and Jan Nieuwenhuizen. LilyPond, a System for Automated Music Engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, volume 1, pages 167–171, Florence, 2003. Citeseer.

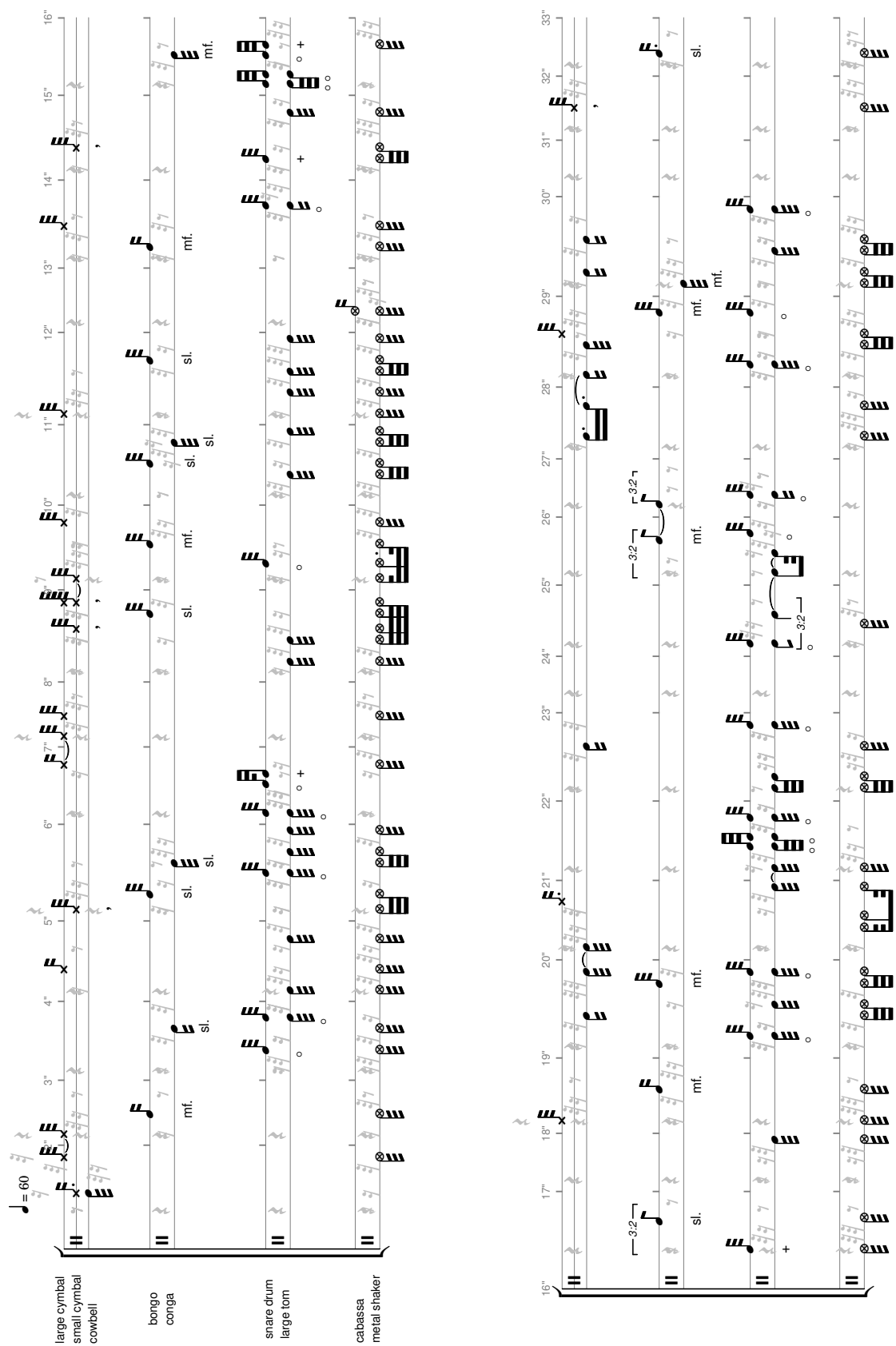


Figure 3: Abjad/LilyPond sketch with the raw orchestration of the target sound using the technomorphic model of computer-aided composition.

grainBirds: Evolutionary granular synthesizers distributed in Fog Computing

Jônatas Manzolli^{12*}, Edelson Henrique Constantino²

¹Arts Institute – University of Campinas
Rua Elis Regina, 50 – Cidade Universitária “Zeferino Vaz”, Barão Geraldo, Campinas, SP

²Interdisciplinary Nucleus of Sound Communication - University of Campinas
Rua da Reitoria, 165 – Cidade Universitária “Zeferino Vaz”, Barão Geraldo, Campinas, SP

jonatas@nics.unicamp.br, edelson@nics.unicamp.br

Abstract. The article presents the grainBIRD concept inspired by a distributed fog of grain sounds metaphor. Shortly, granular synthesizers are distributed in a mobile device’s network. Applying Interactive Genetic Algorithm (IGA) and exchanging Open Sound Control (OSC) messages in a Virtual Private Network, grainBIRDS generate sound grains in standalone and network configurations. The grainBIRD’s concept dialogues with the Internet of Things (IoT) paradigm since it was inspired by Cloud and Fog computing architectures. In this sense, this article presents the grainBIRD and grainBIRD Orchestra concepts, followed by a review on Computer Music and Network technologies related to the project. The article also discusses the computer implementation of the grainBIRD application using the Pure Data programming environment and concludes with performance tests of the network communication feasibility and the system sound generation capacities.

1. Introduction

Given the challenge of creating and interacting with other artists remotely caused by the COVID-19 pandemic, grainBIRD was created. It is possible to say that the grainBIRDS establish a network to share sound control parameters from one place to another: the flighting of messenger birds. The grainBIRD’s concept was born in a series of musical letters called *cartas@todocanto*. Letters were sent to musicians and dancers [1] in different geographic locations, in Brazil and abroad. Shortly, grainBIRDS carry parametric sound letters whose exchange among devices generates a fog of sound grains. The network interaction was projected as a suitable process for distant co-creation among musicians and dancers. Particularly, the use of the internet as new environment offering to composers and creators a medium for collective creation and production of open and continuously evolving works, as discussed by [2].

The implementation of grainBIRD brought together different techniques from the Computer Music literature such as the contemporary notion of laptop orchestras [3, 4, 5] and also the Ubiquitous Music perspective [6]; the granular synthesis [7], [8] used to implement grainBIRD’s synthesis engine. Further, the mobile granular synthesizers operate with Interactive Genetic Algorithms (IGA) [9, 10, 11, 12]. grainBIRDS communicate in two

ways: a) with each other in a network of mobile devices, and b) with desktop host computers. Thus, grainBIRD network implementation dialogues with the Internet of Things (IoT) paradigm [13], Cloud Computing [14][15], and Fog Computing [16].

Communication among grainBIRDS is carried out by exchanging Open Sound Control (OSC) messages [17] between local devices, such as the fog infrastructure concept, and over the Internet through the Virtual Private Network (VPN) [18] interface. Considering that all devices will communicate locally or remotely, OSC messages are exchanged through a multicast group. Figure 1 introduces the network communication architecture in line with the following concepts:

grainBIRD: application for a mobile device that runs a granular synthesizer controlled by IGAs. The synthesis is controlled locally or by OSC¹ messages exchanged in a grainNET.

grainBIRD Orchestra or grainNET: Orchestra formed by communicating the different grainBIRDS with each or with a host within a VPN² network.

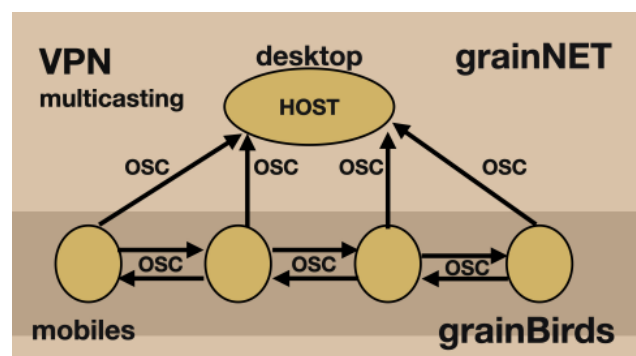


Figure 1: Diagram illustrating the concept of grainBIRDS in a network.

¹<http://opensoundcontrol.org>

²It is a VPN that extends a private network by a public network allowing users to send and receive data over public or shared networks, as if their devices were connected to the private network directly.

*Supported by CAPES.

2. Dialogue with Computer Music Technologies

Networked musical performance might date back to a concert performed on July 3, 1978 at Blind Lemon, a small music gallery in Berkeley, California [19]. It could be the first collective performance of a composition distributed in a microcomputer network. Later, in 2005, the Princeton Laptop Orchestra, an ensemble of fifteen laptop-based meta-instruments, began its first season [5]. The possibility of using distributed systems in many embedded devices available for musical composition and performance dialoguing with IoT technologies, is discussed in [13]. grainBIRD dialogues also with the ubiquitous music perspective [6], in a previous work we described the interaction and collaboration in computer music using networks [14].

2.1. From Laptop Orchestra to grainBIRD Orchestra

A laptop orchestra or laptop ensemble is a chamber music ensemble consisting primarily of the collective use of laptops as musical instruments. With artistic and educational purposes, these orchestras focus on creating innovative technologies and artworks that use improvisation and frequent collaborations with other ensembles and soloists, dancers, video artists, and actors [4].

In the study presented here, we implemented a derivation of the laptop orchestra concept. Besides, the grainBIRDS provide means for local performances, distributed performance with mobile devices, or performances merging local ensembles with virtual ones.

2.2. From Granular Synthesis to Grain Fog

Granular synthesis (GS) was born with Dennis Gabor's concept of a "sound quantum" in 1947 [20]. Nowadays, it is defined as a digital synthesis operating in the time scale of microsounds. It is based on sequencing samples that are partitioned into small segments called "grains." Specifically, these grains have durations less than tenths of a second and great than tens of milliseconds [7].

In the project discussed here, we draw an analogy between fog computing and the fog of granular sounds generated by mobile devices. A grainBIRD contains sound samples modulated by envelope curves and parameters such as the start reading point, duration, pitch, and overlapping rate of each grain. These five parameters are controlled by IGA, as presented next.

2.3. Interactive Genetic Algorithm in a Network

Digital sound synthesis is a natural domain for applying evolutionary computation [11]. The concepts of the genome and genetic operations can be consistently mapped into synthesis parameters [11]. In previous work, we have already introduced the concept of evolutionary synthesis [10]. Another argument is that GS can give rise to unexpected and complex changes in the sound output, and with Genetic Algorithms (GA), it could be possible to deal with that within a semi-automatic approach, using human intervention. The IGA is also a general term for GA that uses

human evaluation as a fitness function [9]. In other words, the synthesis process follows the user's particular preference. We have also discussed IGA with the concept of ArTbitration [15], which denotes the users' aesthetic judgment using evolutionary computing. A way to interact with sound processes and increase the complexity of the resulting sound without neglecting aesthetic aspects.

The application of IGA was relevant for our project because grainBIRDS are controlled explicitly from the graphical interface and implicitly from motion sensors. The coupling between GS and IGA provides a fertile field for exploration. Digital sounds would be generated with vitality and musical interest in a wide range of temporal scales and timbres.

3. Dialogue with Network Technologies

The IoT concept is guided by creating a dynamic network infrastructure to connect anything, which can be physical or virtual objects, through different means of communication in which each device receives a unique IP address. Therefore, the devices have characteristics such as identity, physical attributes, positioning sensors, monitoring, and other sensors that collect some type of information. Such information is further communicated and controlled through the Internet [21]. In the grainBIRD's network concept, the access through a VPN and the OSC protocol enables the morphology control of a distributed fog of sound grains.

3.1. Internet of Things

A grainBIRD comprises what we define in IoT as devices that perform one or more operations within a specific or dedicated scope. The grainBIRDS communicate and exchange messages with the other devices present in the communication infrastructure through IP and OSC protocols.

In order to allow efficient communication among grainBIRDS, the established and response time (latency) among the devices must be in the desired minimum value (see tests performed below). When executed in local infrastructure, time could not be a limiting factor. However, as in grainBIRDS, time latency becomes a significant challenge when it operates in a distributed infrastructure.

Given the above, we experimented with cloud and fog computing paradigms to create a high-performance digital infrastructure to enable connection and communication between grainBIRD devices. We looked for local and remote latency that could make it possible to exchange granular synthesis parameters, trigger, and other specific controls.

3.2. Cloud Computing

Cloud computing enables the development of solutions and technologies without significant hardware or skilled labour to support and maintain services infrastructure [22]. Additionally, Cloud Computing offers agility in the supply of hardware, better use of computing resources, and the possibility of redefining resources at any time, enabling the

development of projects requiring high performance and dynamic resource allocation.

The National Institute of Standards and Technology (NIST) defines Cloud Computing as “a model that enables ubiquitous, convenient and on-demand access to a set of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be quickly provisioned and released with minimal management effort or service provider interaction” [23]. In this context, the computational resources are made available and accessed through the Internet from the existing network infrastructure on site [24].

In order to meet the communication requirements between grainBIRD devices, local and remote, we explored the cloud and fog computing paradigms, which has, among its characteristics, the expansion of the local network coverage infrastructure to ensure stability and high performance, such as low latency, as required for the performers’ interaction and the sound synthesis control shared among grainBIRDS. The fog computing concept is presented next.

3.3. Fog Computing

Applications that require low latency, high availability, or mobility can be hampered in locations or regions where coverage and bandwidth are limited or even non-existent. The fog computing paradigm is linked to the implementation of cloud computing resources at the edge of network infrastructures, such as data storage or processing. According to Yi et al. [25], fog computing provides dynamic resources and services to users at the edge of the network, while cloud computing provides resources distributed across the core network.

For Nandyala and Kim [26], fog computing can “act as a bridge between smart devices and large-scale cloud computing and storage services.” In the present project, we define fog computing as a decentralized architecture in which grainBIRD devices provide information and communicate through the digital infrastructure built between the network edge and the Cloud.

As shown in Figure 2, grainBIRD devices communicate locally and remotely via the grainNET network infrastructure (i.e., the grainBIRD Orchestra). This network is based on IoT – devices connected locally that compose a haze for communicating with each other via cloud computing. In this setting, they communicate and interact with other remote devices when connected to the grainNET network.

In short, grainNET is a digital infrastructure that permeates the connection between various grainBIRDS’ fog – which is integrated into the device cloud. We understand the sound generated by the grainNET in two ways: 1) a texture generated locally by the grainBIRDS and 2) the texture generated in a host computer by the sum of the OSC messages send by grainBIRD devices in a VPN interface and multicasting manner.

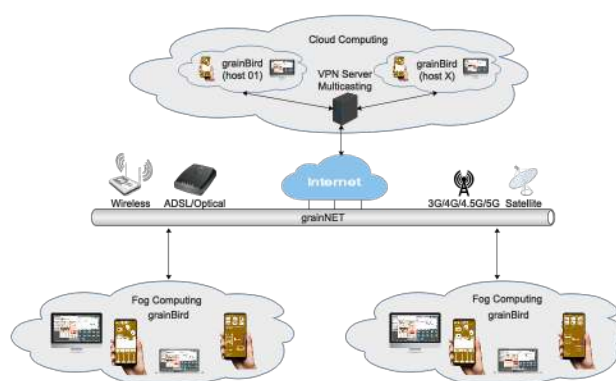


Figure 2: Digital infrastructure for communication between the Cloud and Fog among grainBIRDS.

4. grainBIRD Implementation

The following paragraphs discuss the features and architecture of the grainBIRD implementation with Pure Data Vanilla³ programming environment [16] and freeware MobMuPlat, created by Daniel Iglesias⁴. The grainBIRD’s granular synthesizer was derived from Farnell [27].

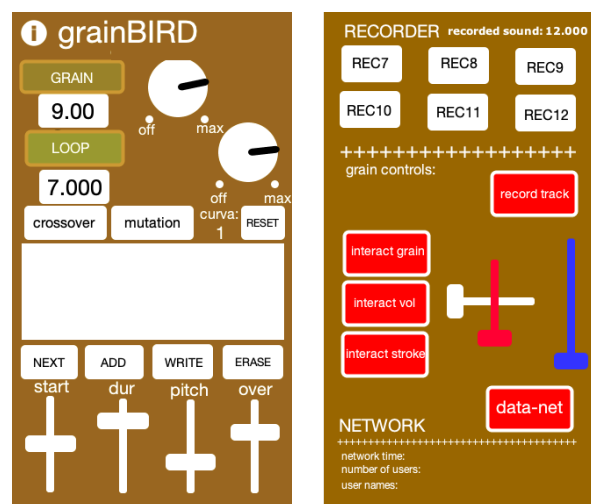


Figure 3: GrainBIRD GUI: granular synthesizer controllers (left) and toggles to control local recording and network communication (right).

4.1. grainBIRD application

The grainBIRD App allows the generation and transformation of sound grains. It is explicitly controlled with the GUI (see Figure 3) and implicitly by motion sensors. Using Pure Data programming language, we developed the grainBIRD App organized in four main groups of sub-patches: 1) to play and generate sounds, 2) to execute IGA operations, 3) to store synthesis pre-sets, and 4) to control network communication. Figure 4 presents the grainBIRD App main patch.

³<https://puredata.info/downloads/pure-data>

⁴<https://danieliglesia.com/mobmuplat>

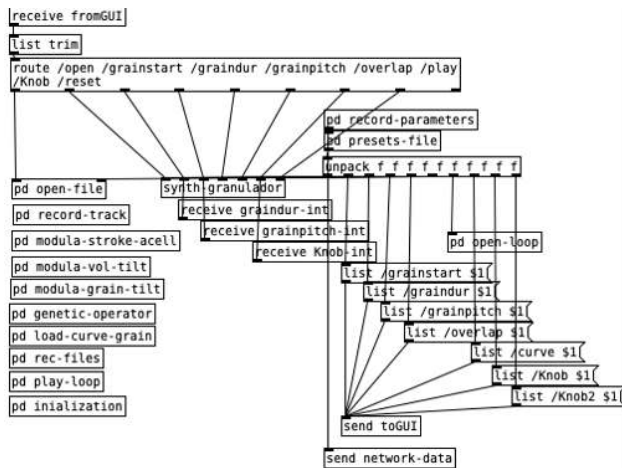


Figure 4: Main Patch of grainBIRD App.

4.2. Interactive Evolutionary Granular Synthesizer

The IGA implementation comprises a population, a genome, and two genetic operators: Crossover and Mutation (see [12] for more explanation).

PARAMETER	RANGE	UNIT
<i>gen. operation</i>	[-1,0,1]	
<i>grain start</i>	[0..1]	%
<i>grain duration</i>	[50..2000]	milliseconds
<i>grain pitch</i>	[0.1..2]	freq. ratio
<i>grain overlap</i>	[1..2]	
<i># Sound grain</i>	[1..12]	
<i># Sound loop</i>	[1..12]	
<i># Envelope</i>	[1..12]	
<i>grain volume</i>	[0..100]	dB
<i>loop volume</i>	[0..100]	dB

Table 1: Parameters of the synthesis genome.

As described in Table 1, a 10-parameters list controls the granular synthesizer, the number indexes of sound grain, sound loop, envelope, and the volume of sound grains and sound loops. The IGA control of grainBIRDS follows this idea: mutation and crossover operations are controlled by local and remote players acting on the GUI or producing movement with their mobile phones. Thus, motion sensors, the accelerometer and tilt sensors, produce changes in the granular flow.

Another GS control is related to the shape of grain envelopes defined originally here as a 2048-points Hamming window, see implementation [27]. Moreover, the shape of these envelopes is also transformed by IGA in the following ways: 1) drawing the envelope on the GUI, 2) clicking on the crossover or mutation buttons to draw curves automatically, 3) mixing both.

Secondly, it is also possible to transform the envelopes by moving the mobile phone in two ways. The first one is associated with the Tilt sensor. When the user roll or yaw the mobile phone, the synthesis slider controls (see Figure 3) move together. Secondly, the crossover operation is triggered when the acceleration module related to the user's movement exceeds a threshold. Figure 5 displays two columns of three envelopes describing an automatic evolution starting upon the Hamming window and subsequently applying a crossover operation.

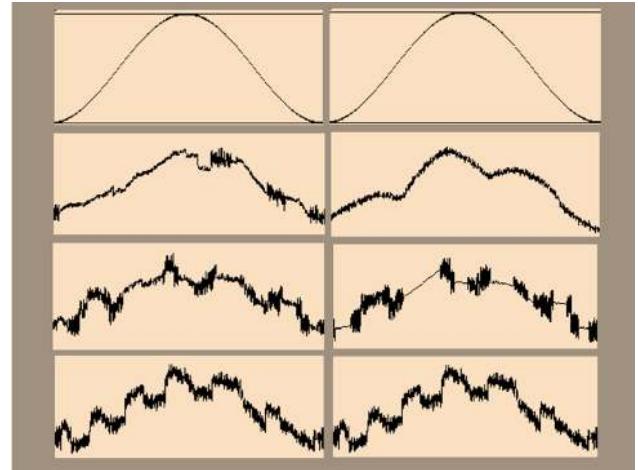


Figure 5: Automatic Evolution of grainBIRD envelopes from top to down: a hamming window is followed by 03 generations of envelopes.

The two types of evolutionary granular synthesis controls (which are composed of ten parameters and the envelope shapes) are stored in two files. Following the Evolutionary Computing paradigm, they are considered here as the grainBIRD genomes [11]. The first group, consisting of 12 files, contains the numerical values of the envelopes. The second file stores the 10-parameter list in each line as the user adds it to the file. In Figure 3, it is possible to see four buttons: Next, Add, Write, and Erase to control the file storage.

Finally, twelve sound samples are used to generate the grains, and the play sound loops are stored in WAV format of 16 bits and 44.100 Hz – the original grainBIRD implementation contains samples of Brazilian bird songs. It can be renewed in two ways: 1) storage a new group of twelve samples in the /sounds folder, and 2) six sounds can be recorded by the grainBIRD's app itself during a performance.

4.3. Network Communication

The flow of OSC messages [17] generated by the grainBIRDS enables interaction between artists in different remote locations. Each device produces its particular sound fog; they coexist in time and could be heard simultaneously or not. The system does not transmit audio in real-time, but only OSC messages among the network. Nevertheless, a performance with grainBIRD can be transmitted by a network audio distribution software enabling the coexistence of sounds in a synchronized texture.

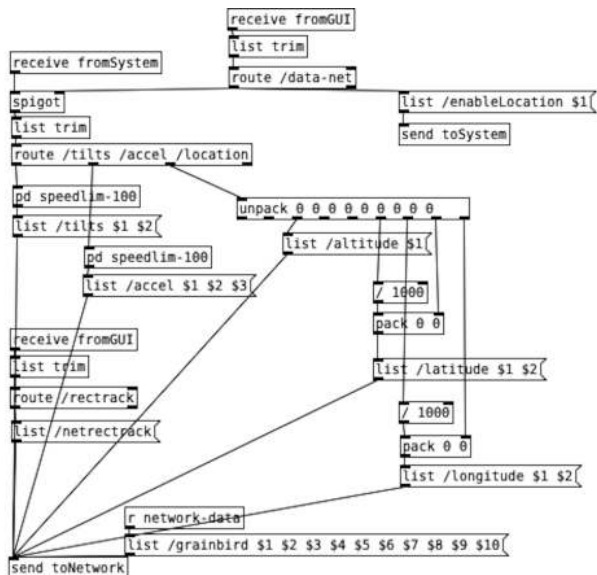


Figure 6: Sub-patch describes the grainBrain OSC communication implementation.

As described previously, the grainBIRDS were implemented as local standalone synthesizers whose parameters are modified by an IGA locally and by OSC messages remotely. The 10-parameter control list is transformed into OSC messages, and they are propagated in a VPN multicasting manner [18]. In addition, each grain-BIRD transmits the variation of the motion sensor (Tilt) and the accelerometer, altitude, and geo-location. This data is used on the host grainNET machine to control the granular synthesis automatically. We will also implement a graphic plot of the geo-distribution of the grainBIRD Orchestra when the performers are performing from different remote locations. Figure 7 shows the GUI of the grainNET host computer, which is a mirror of the grainBIRD App.

5. Performance Tests

We carried out some tests of grainBIRD's network latency using a VPN and sound capacities – to measure its feasibility in real-time performances. In this setting, standalone solo and distributed performances were evaluated.

5.1. Network Performance

The VPN [18] used for testing can be described as a logical network created on top of the existing private or public physical infrastructure to provide a logical digital infrastructure for equivalent communication. This logical communication infrastructure was established from grain-BIRD device(s) to the central VPN server, which is installed, in our experiments, at the Interdisciplinary Nucleus of Sound Communication (NICS) at the University of Campinas (UNICAMP). The VPN system is part of the set of services and applications allocated in cloud computing, as shown in Figure 8. In our implementation, the VPN uses public-key cryptography (asymmetric) with combined 2048-bit RSA keys with digital certificates signed by the server, making it highly secure and reliable. In this format,

keys are initially generated on the server and associated with grainBIRDS devices, guaranteeing integrity, authenticity, and security in connection and communication.

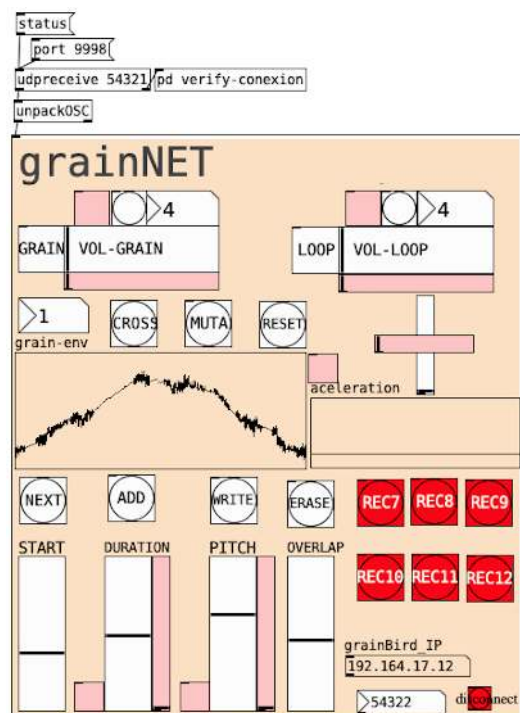


Figure 7: GUI of the grainNET Host computer.

As can be seen in Figure 8, after connecting to the VPN network and receiving the unique IP address, a grain-BIRD starts to receive the messages sent through multicasting automatically. In addition, it was possible to view and communicate with the other grainBIRDS connected to the grainNET network. Therefore, through that bilateral communication, all grainBIRDS can send and receive messages through VPN, which was also defined in this project as a grainNET network (see Figure 2).

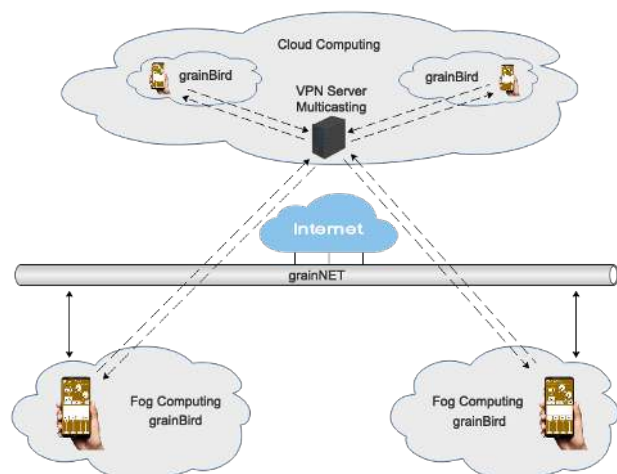


Figure 8: Multicasting communication over the VPN network used for testing the grain-BIRDS.

Two experiments were carried out to measure and evaluate the communication among grainBIRDS: a)

sending 100 packets (echo request/reply) between two local grainBIRD devices, and b) sending 100 packets (echo request/reply) between two grainBIRDs devices over the VPN network. As defined in this approach as fog, both devices were connected to the same network via a wireless router for the local test. For the remote test, one of the devices was connected to the VPN through the 4.5G mobile network provided by the operator Claro BR, and the second device through the home wireless network.

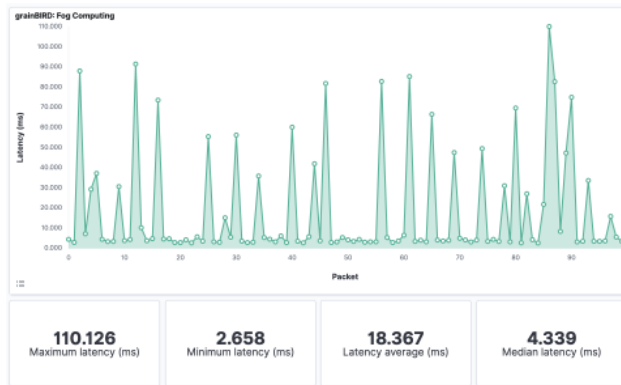


Figure 9: Graphic of the max, average, minimum, and median latency obtained in a local fog network.



Figure 10: Latency max, average, minimum, and median latency obtained via the cloud VPN network.

Figure 9 shows the plot of latency obtained in the communication between two local grainBIRDs, that is, in the fog layer. Figure 10 shows the plot of latency obtained between two remote grainBIRDs, that is, communicating over the VPN network.

5.2. Sound generation performance

Firstly, we tested a solo performance just with one grainBIRD in a standalone situation. Figure 11 shows the audio track produced by the evolution of five envelopes while it was applied only to crossover operations.

Figure 11 also displays a sequence of five grainBIRD envelopes (in the top) align with the spectrogram (in the bottom). The vertical lines indicate changes on grain envelopes between approximately 20 seconds. Thus, it is possible to see the spectral content of the sound generated

by that grainBIRD solo. It is important to note that the envelopes at the top of Figure 11 are applied to many grains of the granular synthesis (see Figure 5).

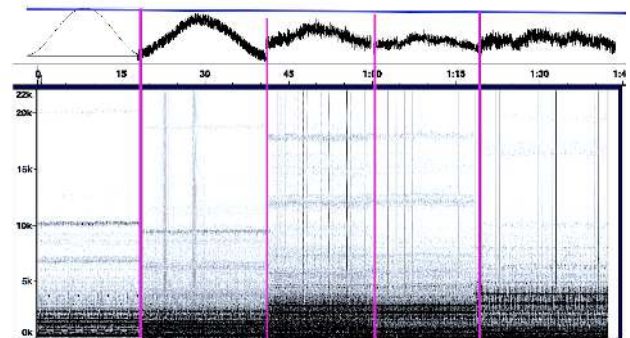


Figure 11: Recorded audio track during the grainBIRD solo test.

Secondly, we carried out a distributed performance test with two grainBIRDs and a host grainNET computer from a distance in a VPN environment. The grainBIRD devices and the host computer were synchronized with a start-play OSC message to record local soundtracks simultaneously. After that, OSC messages were sent over the network from a guiding grainBIRD so that the others responded together. It was sent from the grainBIRD1 05 crossover messages.

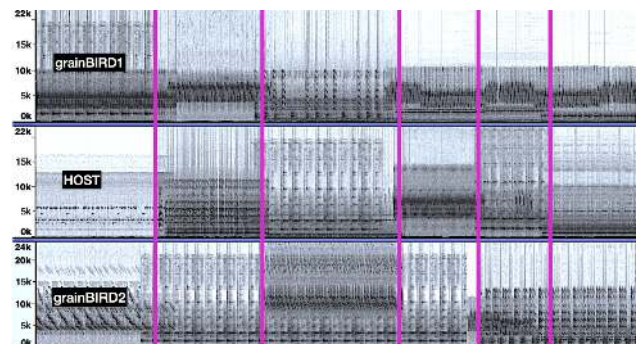


Figure 12: Spectrogram of the grainBIRD distributed performance.

In Figure 12, the local recordings of each mobile device are displayed together in three superimposed spectrograms. The top plot is from a grainBIRD1 run in an iPhone, the middle is from the Host grainNET run in a MacBook desktop, and the bottom is from a grainBIRD2 run in an iPad. The vertical lines indicate the grain envelopes synchronized changes over the network.

6. Discussions

From the graphics presented in Figures 9 and 10, it is possible to evaluate the latency range produced in a test exchanging 100 messages between two grainBIRD devices. Therefore, regarding the grainBIRD connection test, the latency obtained in the communication between the devices was an average of 18.367 ms, which is good network performance, and it meets the requirements of the grainBIRD devices. On the other hand, the latency obtained

in the communication between remote devices during the tests through Cloud was on average 98.008 ms. This result was considered promising and allowed us to control and exchange messages between existing grainBIRDS in a remote fog network.

Regarding the solo performance test, it is possible to verify that the automatic evolution of grainBIRD envelopes can generate and transform the content of sound grains. The granular synthesizer of each grainBIRD already produces its sound fog, which is then associated with other grainBIRDS, thus forming a complex texture, as shown in Figure 11.

From the sound distributed performance test, the synchrony and complexity of the resulting soundtracks can be visualized (see Figure 12). The guiding device was used as a kind of orchestra director for grainBIRDS performed synchronized sound transformations. The network latency was not significant for generating sound grains as it happened quite simultaneously. It is crucial to notice that even there were transmission delays on sending the 10-parameters messages (see Table 1). This list controls granular synthesis macro while micro-controls are performed by each engine locally as defined above (see section 3.1 – grainBIRDS perform operations within a specific or dedicated scope).

Finally, to demonstrate the interaction with mobile devices motion sensors, we develop an improvisation test with three grainBIRDS and a host grainNET computer (see Figure 13). We use four different devices (iPhone, iPad, Moto G3) and a MacBook to test also compatibility between the App, different motion sensors, and OS of these different devices. The test consisted of two users in different remote locations over the Cloud, performing a movement with a device followed by the user in the other location and so on. The result presented in Figure 13 shows that the changes were always propagated through the network evenly and with a slight delay (i.e., see the sync demarcation of the vertical lines).

From the discussion here, it is possible to understand grainBIRD allows multiple performance configurations that can be hybridized using local and remote devices over a cloud or fog infra-structure and can also be part of several distributed performance configurations.

7. Conclusion

Given the social isolation caused by COVID-19 and motivated by the challenge of creating and interacting with other artists remotely, we created the concept of grainBIRD that was proof here. We imagine a haze of mobile devices generating granular sounds via OSC messages within a network. We implement a distributed granular synthesis process using a VPN network and IGA to interact between performers in remote locations. Thus, grainBIRD devices communicate and interact with others using remote devices connected to the grainNET network. Network communication was discussed here in light of the IoT, Cloud computing, and Fog computing paradigms. We

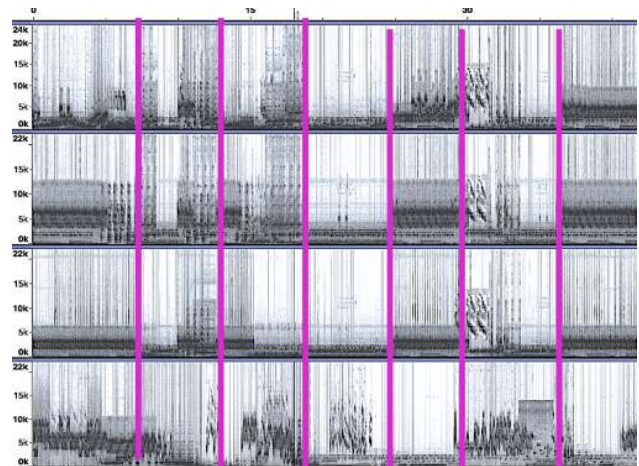


Figure 13: Spectrogram of the grainBIRD distributed improvisation with movement sensors.

showed that the access through a VPN and the OSC protocol enabled the control of the morphology of the sound fog generated by grainBIRDS.

The concept of grainBIRDS is derived from the concept of the laptop orchestra. However, we design the grainBIRDS' app with the notion of cloud and fog computing architecture, enabling both local interactions and performances over a network of mobile devices. In this sense the research presented here dialogues with the possibility of co-creation over the Internet [2] and also with the paradigm of Ubiquitous Music [6]. Therefore, we intend to use the geo-location of grainBIRD's devices to draw a distributed Orchestra map and control other sound parameters. We intend to develop also a communication application between the grainBIRDS and other digital musical instruments, thus providing what we could name as a network-distributed mixed music performance.

Finally, grainBIRD will be fundamental for developing the artistic research involving musicians and dancers, which motivated the development reported here.

Acknowledgment

We thank the CAPES-Print Unicamp project "Poéticas do som e do corpo no diálogo interdisciplinar entre arte e tecnologia: estudo de pesquisa de criação em Humanidades Digitais", 88881.311276/2018-01 and the Universal CNPq project, 429620/2018-7. Manzolli is also supported by the CNPq Pq grant 304431/20184.

References

- [1] Jônatas Manzolli, Mariana Baruco Machado Andraus, and Milena Pereira dos Santos. Entre voos e quedas: (sobre)voos de pássaros (des)cuidados. *XV Simpósio Internacional de Cognição e Artes Musicais (SIMCAM)*, 2021.
- [2] Sergi Jordà. Faust music on line: An approach to real-time collective composition on the internet. *Leonardo Music Journal*, 9:5–12, 12 1999.
- [3] Jason Freeman and Akito Van Troyer. Collaborative Textual Improvisation in a Laptop Ensemble. *Computer Music Journal*, 35(2):8–21, 06 2011.

- [4] Why a laptop orchestra? *Organised Sound*, 12(2):171–179, 2007.
- [5] Ge Wang, Dan Trueman, Scott Smallwood, and Perry R. Cook. The Laptop Orchestra as Classroom. *Computer Music Journal*, 32(1):26–37, 03 2008.
- [6] Damián Keller, Victor Lazzarini, and Marcelo Pimenta. *Ubiquitous Music*. 01 2014.
- [7] Curtis Roads. *Microsound*, volume 148. The MIT Press, 2001.
- [8] Barry Truax. Real-time granular synthesis with a digital signal processor. *Computer Music Journal*, 12(2):14–26, 1988.
- [9] A. Kosorukoff. Human based genetic algorithm. In *2001 IEEE International Conference on Systems, Man and Cybernetics. e-Systems and e-Man for Cybernetics in Cyberspace (Cat.No.01CH37236)*, volume 5, pages 3464–3469 vol.5, 2001.
- [10] Jônatas Manzoli, Adolfo Maia, Jose Fornari, and Furio Damiani. The evolutionary sound synthesis method. In *Proceedings of the Ninth ACM International Conference on Multimedia, MULTIMEDIA '01*, page 585–587, Ottawa, Canada, 2001. Association for Computing Machinery.
- [11] James McDermott, Niall J. L. Griffith, and Michael O'Neill. *Evolutionary Computation Applied to Sound Synthesis*, pages 81–101. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
- [12] Artemis Moroni, Jônatas Manzoli, Fernando Von Zuben, and Ricardo Gudwin. Vox Populi: An Interactive Evolutionary System for Algorithmic Music Composition. *Leonardo Music Journal*, 10:49–54, 12 2000.
- [13] Angelo Fraietta, Oliver Bown, Sam Ferguson, Sam Gillespie, and Liam Bray. Rapid Composition for Networked Devices: HappyBrackets. *Computer Music Journal*, 43(2-3):89–108, 06 2020.
- [14] Flávio Luiz Schiavoni, Pedro H. de Faria, and Jônatas Manzoli. Interaction and collaboration in computer music using computer networks: An ubimus perspective. *Journal of New Music Research*, 48(4):316–330, 2019.
- [15] Artemis Moroni, Fernando Von Zuben, and Jônatas Manzoli. ArTbitration: Human-Machine Interaction in Artistic Domains. *Leonardo*, 35(2):185–188, 04 2002.
- [16] Miller Puckette. Pure data: another integrated computer music environment. In *in Proceedings, International Computer Music Conference*, pages 37–41, 1997.
- [17] Dave Phillips. *An Introduction To OSC*, 2008.
- [18] Ayhan Erdoğan and Dz Yzb. Virtual private networks (vpns): A survey. 01 2008.
- [19] John Bischoff, Rich Gold, and Jim Horton. Music for an interactive network of microcomputers. *Computer Music Journal*, 2(3):24–29, 1978.
- [20] D. Gabor. Acoustical quanta and the theory of hearing. *Nature*, 159:591–594, 1947.
- [21] M Divyashree and H G Rangaraju. Internet of things (iot): A survey. In *2018 International Conference on Networking, Embedded and Wireless Systems (ICNEWS)*, pages 1–6, 2018.
- [22] Michael Armbrust, Armando Fox, Rean Griffith, Anthony Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53:50–58, 04 2010.
- [23] Tim Mell and Peter Grance. NIST Definition of Cloud Computing. *National Institute of Standards and Technology*, 2011.
- [24] Flávio Sousa, Leonardo Moreira, and Javam Machado. *Computação em nuvem: Conceitos, tecnologias, aplicações e desafios*. 2009.
- [25] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: Concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data, Mobidata '15*, page 37–42, Hangzhou, China, 2015. Association for Computing Machinery.
- [26] Sukanya Nandyala and Haeng-Kon Kim. From cloud to fog and iot-based real-time u-healthcare monitoring for smart homes and hospitals. *International Journal of Smart Home*, 10:187–196, 02 2016.
- [27] Andy Farnell. *Designing Sound*. The MIT Press, 2010.

Correlating Big Five Primary Personality Dimensions with Musical Preferences

Luiz Delando Santos Moreira Júnior^{1*}, Guilherme Ribeiro Eulálio Cabral² Giordano Ribeiro Eulálio Cabral²

¹Centro de Informática – Universidade Federal de Pernambuco
Av. Jorn. Aníbal Fernandes, s/n - Cidade Universitária, Recife - PE, 50740-560

²Univasf Universidade Federal do Vale do São Francisco)
Av. José de Sá Maniçoba, s/n - Centro, Petrolina - PE, 56304-205

ldsmj@cin.ufpe.br, grecabral@gmail.com, grec@cin.ufpe.br

Abstract. According to [1] in 2019 an estimated 2.95 billion people were using social media worldwide, with a projection of 3.43 billion in 2023. Social networks increasingly play a fundamental role in the way of expressing ourselves and relating to others, in which it is directly linked to our personality. Several studies relate a person's personality to the most diverse areas, such as predicting political orientation, classifying gender, or even measuring the level of success at work. [2] proposes an algorithm that can allegedly estimate the personality of a Facebook user better than his partner, based on his digital footprint, thanks to AI techniques and the massive number of interactions that users perform on their social media. The focus of this work is to relate the "Big Five" personality theory with musical preferences. Since social media posts register moments and impressions of users' lives, and since music streaming platforms provide material about musical taste [3], we used tools to extract the big five primary personality dimensions from Facebook and audio features from the Spotify Web API, and looked for correlations between musical preferences and personality. Results show some correlations and anticorrelations found, pointing to further research with a broader range of features.

1. Introduction

The massification of social networks are in line with the growing interest of researchers in relating their use with personality theories. One in particular, known as the Big Five [4], has been showing interesting results about the correlations found between personality and musical preferences. For example, according to [3] research, positive correlations were identified between blues, jazz, classical, and folk music with the openness personality trait. In the work of [5], another correlation was found between mass music and the extroversion factor. And the two surveys mentioned using manual data collection as methodology, applying questionnaires to determine which values of each user's personality trait to use, and relating them to information about musical preferences also extracted via questionnaires. This way of collecting data motivated me to look for ways to obtain information in an automatic and scalable manner.

According to [6] it is possible to estimate, only based on digital footprints in social networks, an individual's personality traits and characteristics, such as political

orientation and probability of success at work. In this research, we chose to work with music, as it has a prominent place in our lives, according to [3] in a survey conducted by him, music is the most cited activity when the interviewee is questioned for his hobbies and also that musical preferences are considered more significant than food preference.

We built a platform so that it was possible to collect information from likes and Facebook posts through the API to estimate the Big Five personality traits and the information of musical preferences is retrieved from the APIs made available by Spotify.

2. Big Five Personality Traits

According to [7] the Big Five personality traits theory has been growing in psychology as a way of defining the individual's personality, using the most diverse methods. Big Five model had its origin in a series of researches on personality, coming from other theories and personality traits. [8], for a general review, please see [9]. We chose to use this theory as a basis for its acceptance, but also for its simplicity and the availability of digital tools.

As stated by [5], on Big Five there are 5 major traits that characterize human personality. The first of them is openness. According to [10] people who get a high score tend to be creative people and receptive to new ideas and (as the name implies) are more open to experiencing the new. As [9] explains, these individuals are usually referred to as artistic, curious, and imaginative, whereas those with low scores tend to be the opposite, that is, people with conventional thinking, and with a lesser scale of interest.

The second major trait is conscientiousness, according to [10] this factor brings together characteristics that bring the idea of organization, responsibility, and honesty. Individuals who score high on this factor tend to be trustworthy and dedicated workers, those who score low tend to be more impulsive and relaxed. These people can be described as efficient, organized, and responsible [9].

The third major trait is extraversion, which represents traits linked to sociability, as [8] well-defined extraversion is the amount and intensity of interactions between people and activity level. Individuals who score high on this factor tend to be people who prefer to be in the presence of other people, whereas those who have a low score tend to be more reserved. Second [9] adjectives

*Supported by CNPq.

that qualify these people are energetic, communicative, active.

The fourth major trait is agreeableness, according to [10], it can also mean the level of socialization and characterizes the individual's tendency to be socially pleasant and warm. People who score high tend to always seek the best in everyone, some adjectives that represent him are generous, friendly and accommodating [9]

The fifth and final trait is neuroticism, according to [10], this factor represents personality traits that involve positive and negative affect, anxiety, and emotional stability. Those who score high tend to experience negative experiences more intensely. While those with low scores are more emotionally stable. Adjectives for this factor are anxious, unstable, concerned [9].

2.1. Predicting BigFive Personality from digital footprints

The digital footprint is a term used to track the steps that the individual takes on the network, that is, all actions executed by the user using a device connected to the internet, from computers, smartphones and embedded devices [11]. According to [12] these brands are divided into two types, the first is the passive footprint, which represents the actions they perform involuntarily, for example, the search history, or the time that remains on the determined website, records the location and interactions with network infrastructure and we also have a qualified active footprint in this work. It is blocked when we interact directly, usually used when we want to change samples blocked or posted or liked on Facebook, Twitter, and Instagram on the internet.

In this study, we used a tool developed by the Center for Psychometry at the University of Cambridge, which was developed from a rich set of data, without which model was created based on more than 6 million social network profiles. Its objective is to personalize the Internet, transform the data of the digital activity tracked, for example, Facebook likes IDs, Publications written on Facebook, tweets and texts written by the user such as: emails and personal diaries that they can use as input. This information is sent to the service that returns the individual user profile that is formed by the psychological profile that provides information on Big Five personality, intelligence, life satisfaction, political ideology, religious views, sexuality and professions while the demographic profile used by age, gender and the status of your relationship.

As shown by [6], records of user behavior are easily accessible, for example with a list of Facebook Likes, it is possible to automatically extract some personal attributes. The model created is capable of correctly predicting the individual's origin in 95% of cases, whether it is African American or Caucasian and in 93% of cases it correctly classifies between male and female genders, these are cases of dichotomous variables that have a high hit rate, but there are also numerical variables that showed less precision, but this fact does not rule out the existence of correlations that can be verified in obtaining age with a coefficient of $r = 0.75$, in addition to personality traits as

openness $r = 0.43$, extroversion $r = 0.40$ the other personality traits obtained a slightly lower precision $r = 0.17$ to 0.30 .

Even with the accuracy not reaching the values found in the dichotomous variables, studies show that personality judgments made by computer can be more accurate than those made by humans, in a survey of more than 86,000 people who answered personality questionnaires of 100 items from your friends on Facebook demonstrated that the evaluation performed by a computer is more accurate than that performed by your friends. The computer got value for $r = 0.56$ while friends got a coefficient of $r = 0.49$. The results presented show that the average evaluation accuracy of the computer grows as the number of likes increases, in trained models the computer needs 10, 70, 150 and 300 likes to surpass the average evaluation of human judges, respectively, as colleague co-worker, roommate and family member or a spouse. This curve can be seen in figure 2 [2].

3. Platformization of Digital Music

In the early days of the MIR area, studies were almost limited to specialists in the area, where one of the first challenges was the cocktail party. And with the evolution of the area, tools emerged that began to encapsulate tasks such as MFCC, HMM, and Machine Learning techniques, offering scientists a level of abstraction in the development of their studies.[13]

Thinking of a Platformization of Digital Music scenario, companies like Echo Nest are dedicated to offering a musical intelligence service to students, researchers and people interested in the field. based on more than 35 million of songs where it can automatically extract various information from songs that are divided into two sub-categories: One of defined by musicologists which are: tempo, note, tone, signature of tempo volume, among others [14] . And Cultural context features such as textit Energy, Liveness, Speechiness, Acousticness, Danceability and Valence.

In this study, we chose to use cultural characteristics directly related to the way the listener interprets music, in addition to the listener's ease of interpretation on how their personality can interfere in each one. In addition to the accessibility provided by the Spotify Web API. Table 1 shows a list of the characteristics used in this experiment.

4. materials and methods

In this section, we will explain the profile of the research participants and the procedures for developing the experiments.

The research started with the development of the PersonalityMusic-Crawler capable of collecting data from Facebook users invited to test the tool. and sending it to the API Apply Magic Sauce for analysis and then collect data from Spotify's music preferences and store all this data in a non-relational database. Based on this information,

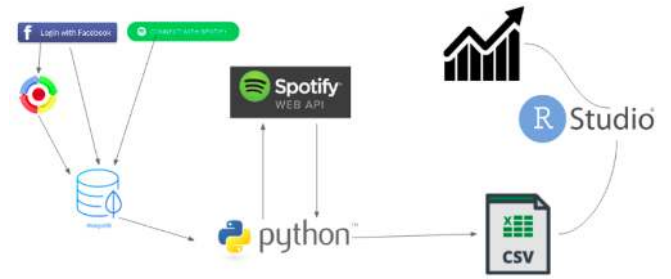


Figure 1: Architecture overview

Features Name	Type	Description
Acousticness	float	A 0.0 to 1.0 confidence measure if the range is acoustic. 1.0 represents high confidence that the track is acoustic.
Danceability	float	Describes how suitable a song is for dancing based on combination of musical elements.
Energy	float	It is a measure of 0.0 to 1.0 and represents a measure perceptual intensity and activity.
Instrumentalness	float	Predict whether a track does not contain vocals. The sounds "Ooh" and "aah" are treated as instrumental in this context.
Liveness	float	Detect an audience presence in the recording. A value above 0.8 provides a strong probability that the track is live.
Tempo	float	The total estimated time of a track in beats per minute (BPM). In musical terminology, Tempo is the speed or rhythm of a given piece and derives directly from the average beat duration.
Valence	float	A measurement from 0.0 to 1.0 describing the musical positivity transmitted by a track. Tracks with high valence sound more positive.

Table 1: Audio features from Spotify Web API

a python script was created to collect the audio characteristics provided by the Spotify Web API and consolidate everything in a CSV spreadsheet so that it was possible to perform the analyzes.

4.1. Participants

About 65 participants were invited to participate in the research as an application tester. 60 records were obtained from the database, 13 of which were incomplete because users do not have enough data to generate any type of recommendation.

On average, each participant had 32 likes to be analyzed, but, by checking the minimum of likes in the sample, it was possible to verify that there were users with 5 valid likes for analysis, which generated a low degree of assertiveness in the classification of the users' personality. As a way of improving, we used only users who had a minimum of 10 valid likes following the example from [2]. In this paper, with 10 selected likes it is already possible to determine the user's personality better than a colleague of work would judge. Another cutoff point was the amount of music that we managed to recover from Spotify for each user. We chose to select only users with at least 100 songs returned by Spotify. This metric was used to identify users who used the Spotify service most often. After these limitations in the database, a total of 31 users were obtained, of which 16 were female and 15 male, aged between 18 and 30 years.

4.2. Instruments

The data was collected by PersonalityMusic-Crawler, which aimed to orchestrate all the following services AMS-Apply Magic Sauce, Spotify Web API, and Mongo DB, which is a non-relational database, chosen for simply allowing new information to be added in the future.

4.2.1. AMS - Apply Magic Sauce

This service is available free of charge for academic or non-profit purposes. We used them to estimate the personality traits from the likes of Facebook, where each user has their Json or a personality element composed of 5 features: BIG5_Openness, BIG5_Conscientiousness, BIG5_Extraversion, BIG5_Agreeableness, BIG5_Neuroticism, and that information was used for the analyzes. Each feature represented as a float ranges from 0.0 to 1.0 and shows personality traits.

4.2.2. Spotify Web API

To retrieve the user music preferences, we use the method "Get a User's Top Artists and Tracks" to collect 100 songs with which the user has more affinity. This request only returns the metadata of the song, such as name, album, band, and popularity. To collect the rest of the necessary information, we used the endpoint "Get Audio Features" service for all tracks that returned the characteristics that can be seen in table 1.

After retrieving the information, the same script was responsible for extracting the average of these characteristics for each user and then exporting that information together with the personality information to a CSV.

The last instrument used was RStudio, had the CSV containing the entire sample with the average of the audio characteristics and personality traits, Pearson's correlation techniques were applied together with a linear regression of the points that obtained the greatest relevance.

4.3. Procedure

In the first contact with the participants, we presented them with the objective of the research, and with his approval, we inserted them as testers of PersonalityMusic-Crawler, to allow Facebook to authorize the recovery of their posts and likes. When the user accesses the application, a short text was available explaining the search.

After the user's login, the application started the information collection process, retrieving all the posts and likes that the user posted on his timeline, sending this data to AMS to be processed, returning the values of the user's personality.

Then he logs in to Spotify and after collecting his music and storing it. A script retrieves all users in the database. Selecting the 100 songs of each user for analysis of the Spotify service.

After recovering the characteristics of each song, the average of these songs is calculated making a unique value that represents the user. So that we could look for correlations between personality traits and musical preferences. This information was processed and stored in a .CSV file along with gender and personality information. With structured data, we used the R statistics software to perform the correlation and regression calculations.

5. Results and Discussion

In this section we will present the results obtained from the statistical analysis between the Big Five factors and the user's musical preferences and raise discussions about the main associations found.

Through the Pearson correlation, some variations were found with the p-value less than 0.05. They are Extraversion vs Danceability, Overture vs Valencia, Openness vs Danceability, and Opening vs Danceability. In table 2, we have a summary of all the association found, where a * marks a p-value less than 0.05 %.

5.0.1. Extraversion vs Popularity

It is worth mentioning a specific case, between Extraversion and Popularity, which obtained marginally positive correlations, since its p-value was very close to 0.05% for an $r = 0.35$; $p\text{-value} = 0.05292$.

We had a $R^2 = 0.09$ we can see in figure 2 the linear regression model extraversionVSpopularity.

$$P = 43.11 + 23.538 * E \quad (1)$$

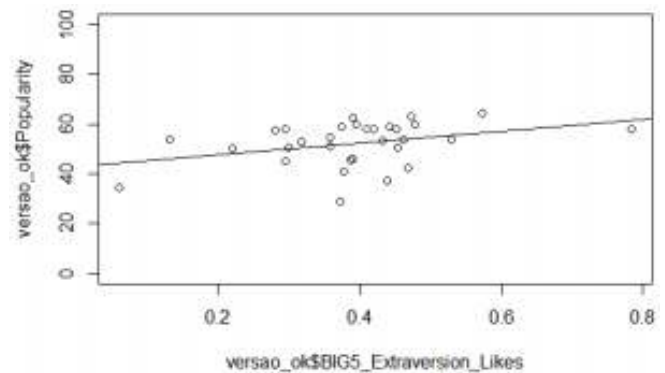


Figure 2: Linear regression - Extraversion vs Popularity

5.0.2. Openness vs Valence

It's found a negative correlation with $r = -0.42$; $p\text{-value} = 0.019$. The value of $R^2 = 0.147$ we can see in figure 3 the linear regression model Openness vs Valence.

$$V = 0.59242 - 0.25512 * O \quad (2)$$

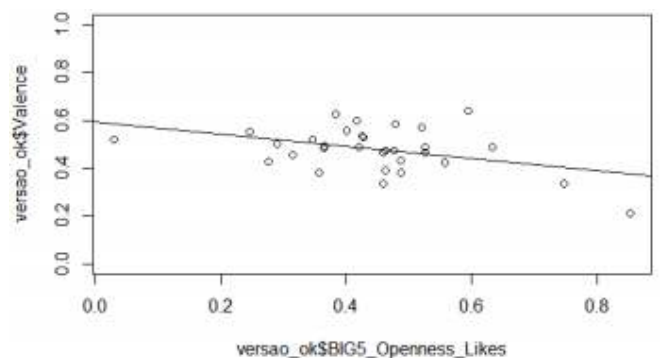


Figure 3: Linear regression - Openness vs Valence

5.0.3. Extraversion vs Danceability

It's Found a positive correlation with $r = 0.40$ $p\text{-value} = 0.02$. The value of $R^2 = 0.13$ was obtained. We can see in figure 4, the linear regression model Extraversion vs Danceability.

$$D = 0.45114 - 0.22402 * E \quad (3)$$

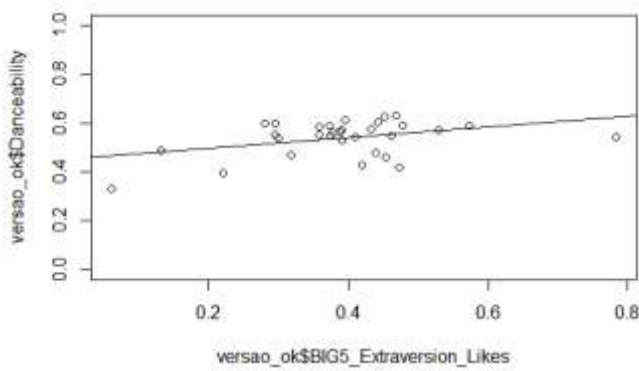


Figure 4: Linear regression - Extraversion vs Danceability

5.0.4. Openness vs Danceability

It's found a negative correlation with $r = -0.38$; $p\text{-value} = 0.37$. The $R^2 = 0.11$. We can see in figure 5, the linear regression model Openness vs Danceability

$$D = 0.61902 - 0.18053 * O \quad (4)$$

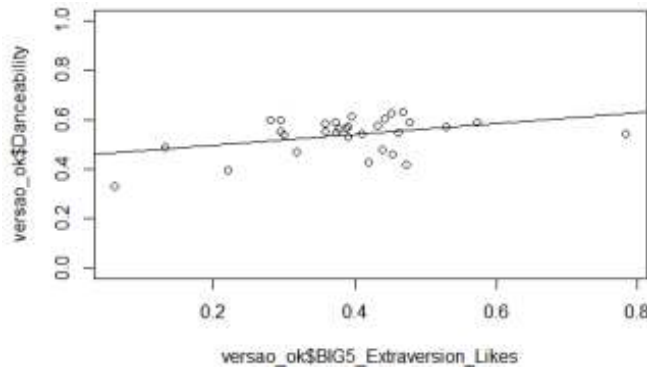


Figure 5: Linear regression - Openness vs Danceability

5.0.5. Agreeableness vs Danceability

It's found positively Association with $r = 0.38$; $p\text{-value} = 0.036$. The $R^2 = 0.11$. We can see in figure 6, the linear regression model Agreeableness vs Danceability

$$D = 0.47286 + 0.14903 * A \quad (5)$$

For the characteristic dancability, three personality traits were found that had a relevant level of correlation. In order to find a model that would bring a better result to describe the dance, we performed multiple linear regressions. As a significant result we had the model using the following variables D = Danceability E = Extroversion, O = Openness, A = Agreement. eqnref danceability

$$D = 0.39 + 0.21 * E + 0.13 * A \quad (6)$$

Neste modelo, foram obtidos os seguintes valores de $p\text{-value}$ respectivamente $p = 0,026$ e $0,037$. e o valor para R^2 newunicodechar ^{2} assecurama $^2 = 0,23$.

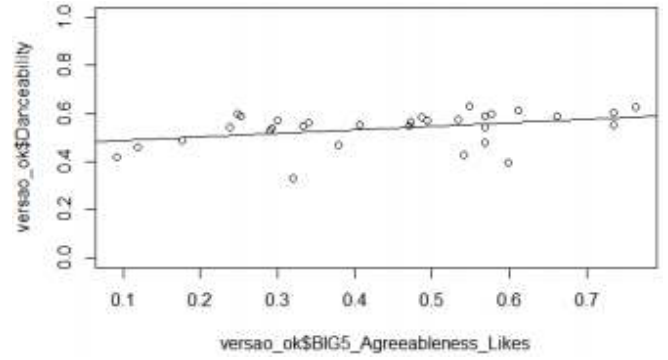


Figure 6: Linear regression - Agreeableness vs Danceability

	Acousticness	Danceability	Energy	instrumentalness
Neuroticism	0.31	0.21	-0.32	-0.20
Conscientiousness	0.02	0.11	0.28	-0.07
Extraversion	-0.30	0.40*	0.20	-0.44*
Openness	0.09	-0.38*	-0.29	0.30
Agreeableness	0.20	0.38*	-0.22	-0.02
	Liveness	tempo	Valence	Popularity
Neuroticism	0.06	0.17	0.09	0.23
Conscientiousness	0.20	0.40*	0.33	0.11
Extraversion	-0.16	0.25	0.35	0.35
Openness	-0.08	-0.22	-0.42*	-0.21
Agreeableness	-0.06	0.05	0.19	0.21

Table 2: Correlation between BigFive traits and Spotify characteristics

5.1. discussions

Of the correlations found, there were three of them that were more relevant and seemed to have an explanation of cause and effect. A marginally significant relationship was found between Popularity and Extroversion, this fact possibly occurred due to the number of users that were in the analysis. In the initial analyzes with fewer users, it was possible to identify this correlation with a $p < 0.05$. This was possibly due to some distortion in the data, but according to [15] the correlations increase with the size (and variability) of the sample. With this, it is expected that with the increase of the sample the value of p decreases bringing greater confidence for the relation.

Comparing with the relationships found in the work of [5] we can see that mass music was positively related to extraversion and negatively to openness to experience, which was also the relationships found with the dance characteristic of the music, because when we observe the genres present in popular music we note that they are all genres that tend to be danceable.

The correlation between openness and Valence of music showed a significantly high negative correlation, we can interpret this correlation as people who have a lower value of opening up to new experiences, that is, more closed individuals in their world tend to listen to music that they like, looking for more positive songs that reinforce your feeling. People with a higher value of Openness tend to be more receptive to new ideas, which ends up leading this user to try and listen to songs that have a low Valence, that is, more negative sounds, exemplifying music (sadder, depressive, angry).

The result of the multiple linear regression showed us that it is possible that several personality traits can influence the characteristics of the songs as was the case with danceability. This leads me to think about what other elements of personality we can relate to music and use an approach based on the individual's personality to create initial recommendations for songs and podcasts.

6. Conclusions

The present work promoted research in the areas of personality theory involving the Big Five being obtained from a user's Facebook digital footprint. In which it is possible through the use of the Apply Magic Sauce application to estimate the 5 major factors based on the likes on Facebook pages and correlate this information with characteristics such as Valence, Popularity, Energy, Tempo, Danceability, Liveness, Instrumentalness extracted from the 100 songs that the user has more affinity with one of the biggest streaming services Spotify.

Based on these analyzes, it was possible to identify three main correlations between Valence and Openness to Experience and two other correlations that were found in our analysis: Extroversion and Popularity, Extroversion, and Danceability. These last two were also found similar relations between extroversion and music of very popular Brazilian music that is represented by songs of the genre: sertaneja, pagode, pop music, funk, forró, samba, axé, and Brega, according to [5] which highlights the correlation between these factors.

7. Future works

Add new entries for data collection, incorporate more data in the correlation analysis, as well as insert other characteristics of the music, both metadata such as band, genre, release year among other information and characteristics extracted directly from the audio such as music note, mood that to enhance analysis such as genres, music notes, and humor. In addition to incorporating an analysis of the lyrics of each song, as we well know that the lyrics have a fundamental role in the emotion that the music passes on to the listener and in this work, we do not take this possibility into account so for future works we can do analysis and keywords, and feeling of the lyrics to understand if there are relations between the music, lyrics and personality factors.

References

- [1] J Clement. Number of global social network users 2010-2021. *Statistica*, 4(7), 2019.
- [2] Wu Youyou, Michal Kosinski, and David Stillwell. Computer-based personality judgments are more accurate than those made by humans. *Proceedings of the National Academy of Sciences*, 112(4):1036–1040, 2015.
- [3] Peter J Rentfrow and Samuel D Gosling. The do re mi's of everyday life: the structure and personality correlates of music preferences. *Journal of personality and social psychology*, 84(6):1236, 2003.
- [4] John M Digman. Higher-order factors of the big five. *Journal of personality and social psychology*, 73(6):1246, 1997.
- [5] Carlos Eduardo Pimentel and Edla Daise Oliveira Porto Donnelly. A relação da preferência musical com os cinco grandes fatores da personalidade. *Psicologia: ciência e profissão*, 28(4):696–713, 2008.
- [6] Michal Kosinski, David Stillwell, and Thore Graepel. Private traits and attributes are predictable from digital records of human behavior. *Proceedings of the National Academy of Sciences*, 110(15):5802–5805, 2013.
- [7] Felicia Pratto, Jim Sidanius, Lisa M Stallworth, and Bertram F Malle. Social dominance orientation: A personality variable predicting social and political attitudes. *Journal of personality and social psychology*, 67(4):741, 1994.
- [8] CHSS Nunes and CS Hutz. O modelo dos cinco grandes fatores de personalidade. *Temas de avaliação psicológica. Campinas: IBAP*, 2002.
- [9] Robert R McCrae and Oliver P John. An introduction to the five-factor model and its applications. *Journal of personality*, 60(2):175–215, 1992.
- [10] Claudio Simon Hutz, Carlos H Nunes, Alice D Silveira, Jovana Serra, Márcia Anton, and Luciane S Wiczorek. O desenvolvimento de marcadores para a avaliação da personalidade no modelo dos cinco grandes fatores. *Psicologia: reflexão e crítica*, 11(2), 1998.
- [11] Alexandre Neves. E-portefólios para a gestão da identidade digital: estudo de caso da plataforma sapo campus escolas. Master's thesis, Universidade de Aveiro, 2011.
- [12] Fabien Girardin, Francesco Calabrese, Filippo Dal Fiore, Carlo Ratti, and Josep Blat. Digital footprinting: Uncovering tourists with user-generated content. *IEEE Pervasive computing*, 7(4), 2008.
- [13] Kyle Robert Arcand II. *Stack Music: Spotify and the Platformization of the Digital Music Commodity*. PhD thesis, Duke University, 2019.
- [14] Jesper Steen Andersen. Using the echo nest's automatically extracted music features for a musicological purpose. In *Cognitive Information Processing (CIP), 2014 4th International Workshop on*, pages 1–6. IEEE, 2014.
- [15] Luiz Pasquali. *Psicometria: teoria dos testes na psicologia e na educação*. Editora Vozes Limitada, 2017.

Motus 4: new developments on a web-deployed software for composing real-time audiovisual interactions.

Augusto Piato Amstalden^{1*}, Tiago Fernandes Tavares^{1,2}

¹School of Electrical and Computer Engineering – University of Campinas
Av. Albert Einstein, 400, Campinas, SP

²Interdisciplinary Nucleus for Sound Studies
Rua da Reitoria, 82, Campinas, SP

a213364@dac.unicamp.br

Abstract. *Camera-based audiovisual interactions have been greatly used in art installations. For such, it is often necessary to use specialized hardware and software. This restricts audiovisual installations to areas that can both afford them financially and be physically accessed by the audience. In this work, we present the development of a web-deployed composition tool for audiovisual interactions that runs on the client side and does not require installing any additional software. The tool allows quickly configuring areas for movement capture, and customizing the corresponding audio feedback. Simultaneously, it provides visual feedback that can aid the audience to understand the experience. Consequently, the tool can be used to compose audiovisual interactions that reach a large audience via web. Moreover, its virtual and ubiquitous nature can foster a diversity of artistic explorations related to the remote characteristic of the relationship between audience and artwork.*

1. Introduction

Art installations are artwork that dialogue with the space around them. Interactive art installations have the additional characteristic of using the audience participation as part of its structure, that is, the artistic context only exists with the participation of an audience [1]. Audience participation has been present in a great amount of artwork throughout history, but only in the last few decades the interaction between the audience and the artwork has been mediated using sensors, actuators, and digital computers.

One important characteristic of digital computers is that they are able to emulate systems that are physically unfeasible. This characteristic is found, for example, in computer games, which commonly have physics engines that are unrealistic, yet foster highly engaging and immersive environments [2]. The link between the physical and the virtual worlds, provided by sensors and actuators, has enabled a diversity of interactive artwork, including augmented performances [3, 4], live audience participation [5], augmented musical instruments [6, 7], instrument development frameworks [8, 9], and art installations [10, 11].

Interactive art installations commonly require specialized equipment, including projectors, loudspeakers, the sensors themselves, computers, and a physical environment to host the installation. These elements can quickly

become prohibitively costly, especially in economically disfavoured areas. Consequently, the access to the artistic richness brought by interactive artwork is socially and economically determined [12, 13].

Moreover, in 2020-2021 the world has suffered the COVID-19 pandemic. Within this period, the access to non-essential locations (including art galleries) has been greatly restricted. Another consequence of the COVID-19 crisis is that the Internet and social media have become more relevant as means to communicate and share information.

Such a worldwide transformation has led to the development of several artwork exploiting virtuality and social isolation [14, 15]. Among those, there are ideas based on audio streams, interactive websites, and online streams of studio-based hardware. These ideas generate requirements for frameworks that allow using real-time client-side audiovisual elements, and a myriad of different frameworks have been employed for such.

This concept aligns with the ideas underlying the MOTUS [10], an audiovisual interactive installation that focuses on using affordable hardware for an engine that captures movement and converts it into musical and visual feedback. Cameras have been used in human-computer interaction for music for decades [3], and MOTUS employs this concept while deploying its software as a webpage, thus only requiring hardware that is becoming increasingly ubiquitous [16, 17]. Therefore, even though it is an art installation, it can be understood as a *virtual* one, as the audience does not have to be physically present in a particular place to participate on it.

In its original conceptualization, MOTUS was devised with intertwined technical, poetic, and aesthetic ideas. However, musicians using MOTUS raised the demand for ways to change the musical and visual structures behind it, that is, to use MOTUS as a music composition tool [10].

Here, we present the fourth iteration of MOTUS, namely MOTUS 4, deployed online at <https://tiagoft.github.io/motus/>. Unlike its predecessors, it does not carry an inherent poetic or aesthetic idea. Rather, it provides a set of tools that allow composers to quickly devise, experiment, and share interaction configurations.

*Supported by CNPq.

The remainder of this article is organized as follows. First, Section 2 brings forward non-artistic applications for the interaction concept discussed in this paper. Then, Section 3 discusses the development steps proposed in this work. After that, Section 4 brings further discussions on possible future applications and developments using MOTUS, and Section 5 concludes the paper.

2. Music Applications

MOTUS is a highly customizable musical composing tool aimed at the video-based interaction with humans. It is inspired in making audiovisual interactive installations [1]. However, this same technology can be used in a diversity of other applications, as we discuss below.

Music has shown to have beneficial effects in Alzheimer patients [18] when used alongside regular therapies. These effects include higher engagement and happiness during treatment, and increased motor coordination and memory strength. Also, rheumatologic disease therapies can use feedback to improve control in daily activities, such as dancing or walking [19]. These two aspects are combined in MOTUS, which provides both a motion-guided feedback and a musical response. Thus, its sound and movement capture engines are potentially useful in these contexts.

Also, nowadays there has been an increase in the usage of video and music sharing social media [20], like TikTok [21] and Kwai [22]. We anticipate that MOTUS' underlying ideas can be used in interesting, valuable filters to transform home-made recordings. Then, the transformed audio-powered videos can be shared with online communities for entertainment, content, or advertisement purposes.

Finally, we highlight that many locations are currently leaning towards sustainable development and becoming smart cities. This transformation calls not only for technical uses of data and sensing, but also specific types of interactive, permanent art exhibitions [23], which aim to improve wellness in urban areas. The MOTUS software is an important step towards making this type of art available even in less wealthy areas, as it uses ubiquitous hardware and it can foster a myriad of aesthetic ideas to be explored by local artists.

All of these applications have guided several transformations to the MOTUS' backend. These transformations aim at facilitating the customization, both by artists and developers. We discuss these technological decisions in the next section.

3. Motus Development

3.1. Technology stack

The first version of MOTUS was implemented using JavaScript vanilla with a Webkit framework, which provides functionalities regarding using the webcam and the

client computer's audio system. This framework has particularities for each webbrowser, and, due to security reasons, its updates frequently change the API. This lead to a difficult maintenance code, and quickly the MOTUS software became malfunctioning and obsolete.

In this implementation, we use another framework, p5.js [24], which provides similar functionalities as those initially provided by Webkit. P5.js, however, has a more stable API, which is maintained by the Processing Foundation [25] and caters to a large community of artists, programmers, and practitioners. P5.js also provides community-based support and a vast collection of example-codes and libraries.

Although p5.js allows for a greater code standardization, there are some browser-specific behaviors that must be accounted for. Different browsers might have diverse security concerns regarding the use of some features like video-cameras, microphones, audio reproduction. However, the current implementation of MOTUS, as far as our tests comprised, is not harmed by these particularities hence the system can be used in any major browser.

The application was deployed online using Github pages (<https://tiagoft.github.io/motus/>). This is especially interesting because it facilitates continuous updates and uses the high up-time provided by the service. Next, we discuss the new features implemented for this development stage.

3.2. New features and variations

The update in MOTUS described here implements new features that arise from decoupling the image processing and audio synthesis ideas from the aesthetic proposals. As seen in Figure 1, the new proposal does not provide an aesthetic proposal of its own; rather, it simply models the flow of information through elements that can be further changed as it becomes necessary.

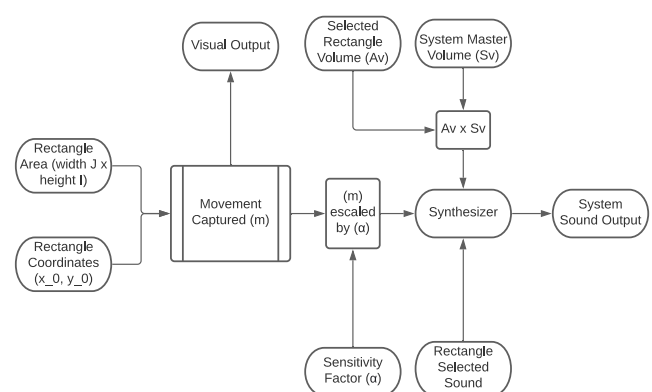


Figure 1: Block diagram for MOTUS' sound engine. Movement is captured from predefined rectangles within the screen and the amount of movement is used as a control parameter to the sound synthesizer.

MOTUS still relies on acquiring movement intensities in rectangles using a webcam video stream within

a web browser, as seen in Figure 2. This concept is important because it allows the software to be deployed via web and to be used with a hardware setup (webcam, laptop speakers, Internet connection) that is common for most computer music enthusiasts.



Figure 2: Screenshot of MOTUS' interface. On the left there are buttons to change general settings and rectangle's settings. On the right there are buttons related to the save and load engine, and to the screen settings.

However, its behavior was changed so composers can create, configure, and delete them. Each rectangle has an area (A), which is defined by its width J and height I , and is placed in the screen at coordinates (x_0, y_0) . The movement within each area is calculated within each $I \times J$ rectangle at each frame t using the grayscale pixel values $p[i, j, t]$, as:

$$m = \frac{1}{IJ} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} |p[x_0+i, y_0+j, t] - p[x_0+i, y_0+j, t-1]|. \quad (1)$$

The movement amount m is highly impacted by characteristics of the webcam and by environmental factors such as ambient lights. Also, because of the averaging factor $\frac{1}{IJ}$, larger rectangles tend to reject more delicate movements. These factors are accounted for by allowing the user to configure a sensitivity factor α that scales m as desired or needed by the composer.

Also, we modified the audio/music synthesis algorithm so that it can be freely changed as to allow artistic explorations. In the current version, users can change the sound sample that is linked to each rectangle, as well as its volume. The composer can use one of the predefined sound samples or upload a sample of their own. The uploaded sample is stored in-browser, that is, it is not stored in any cloud servers, which avoids server liability in any copyright issues that might appear. These features can be seen in Figure 3.

A global setting for the master volume was also added, as seen in Figure 4, allowing a faster on-site configuration. All of these modifications allow the program to be used as a composition tool. This idea contrasts with the earlier 2015 *MOTUS*, in which the whole system was strongly linked to particular aesthetic choices such as the sound production behavior and the motion capture design.

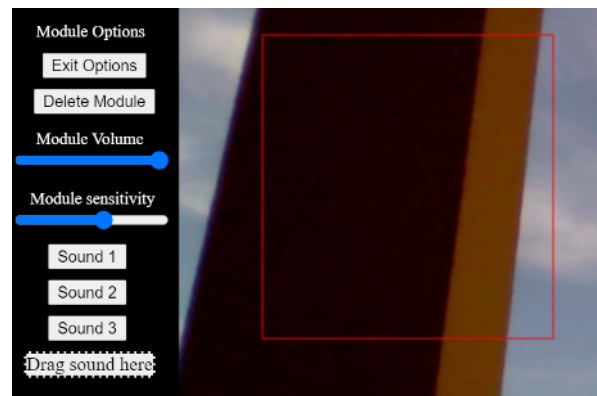


Figure 3: Screenshot crop showing MOTUS' rectangle settings configuration buttons.

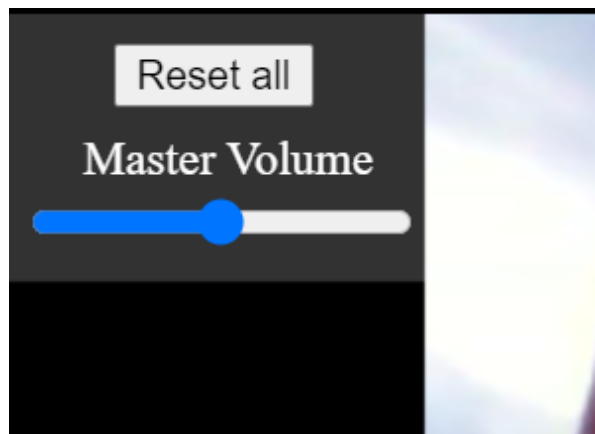


Figure 4: Screenshot crop showing MOTUS' general settings configuration buttons.

Furthermore, new minor components were added to improve usability, like a full-screen mode, a “reset all” functionality (Figure 4) that removes all rectangles, and a red flashing color within each rectangle that lights up proportionally to the calculated movement m serving as visual feedback, as seen in Figure 5. Lastly, we added a save and load option, so that users can export/import their creations as JSON files, allowing them to share their interaction compositions or save them for later use (Figure 6).



Figure 5: Screenshot crop showing MOTUS' visual feedback based on movement. On the left there is a rectangle without movement, on the right there is a rectangle with captured movement emitting visual feedback.

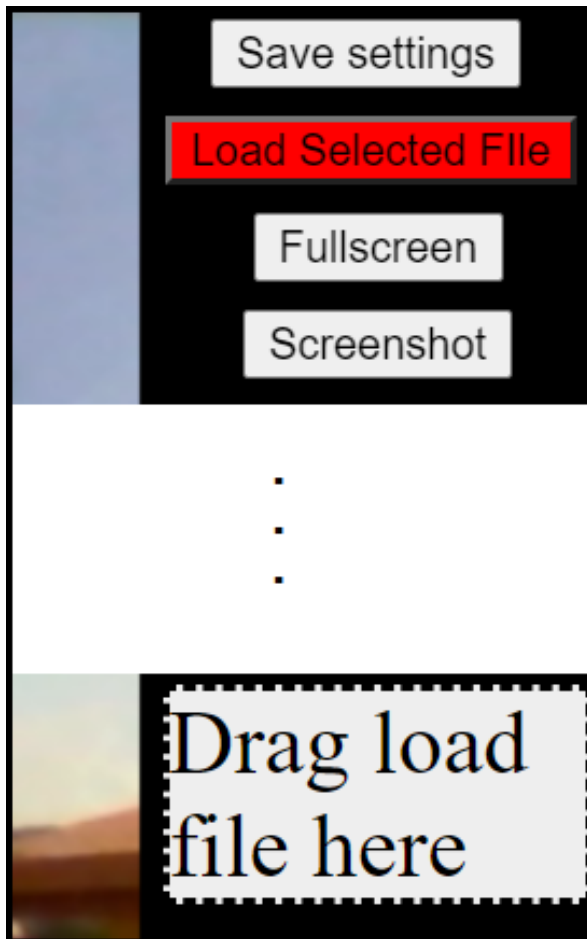


Figure 6: Screenshot crop showing MOTUS' save and load settings configuration buttons and a drag'n'drop area that allows users to upload their own previously saved files.

4. Discussion

In this work, we present a composition software based on a previous audiovisual installation [10]. The composition software untangles the aesthetic and poetic ideas from the technical one, thus allowing quickly composing and customizing a diversity of interactions. It does not require installing specialized software or understanding programming languages, which favors an easier learning process.

As such, we anticipate that composing interactions can be useful in several contexts, many of them diverse from art itself. Table 1 shows some possible applications that can be directly fostered by each of the newly added features.

We highlight that the creative use of each of the features can diverge from our initial idea. As an example, the customizable sound selection feature clearly fits the development of social media pieces and interactive installations, but can play a major role in therapy, as the sound can be changed to enhance the patients engagement.

An important aspect of this composition tool is that it is entirely deployed using the Internet, thus users only have to use a regular web browser. This is important

Features	Applications
Save and Load	Social Media
Customizable Area Selection	Therapeutic Installations
Customizable Sound Selection	Social Media & Interactive Exhibitions
Visual Feedback	Therapeutic Installations
Modifiable Sound Synthesizer	Therapeutic Installations & Interactive Exhibitions
Sensitivity Modification	Therapeutic Installations

Table 1: New features added to MOTUS and their aimed applications.

because it prevents composers from having to install specialized software or learn complicated frameworks. Also, it allows the audience to access artwork without necessarily having to physically go to a specific location. Although not mandatory (as the artist can simply develop an installation and run it on a physical site), the virtuality of this interaction poses a vast field for developments and explorations.

Virtuality in artwork has become especially important during the COVID-19 crisis, which calls for social isolation, hence the use of telecommunication technologies has become necessary. Hence, Motus 4 comes as a timely tool to foster artistic and technical explorations during the pandemics.

5. Conclusion

This paper discusses new technical developments to the MOTUS engine, previously used in an audiovisual interactive art installation. It has been entirely ported to the P5.js framework, which facilitates its development. Also, we have added several new features, as requested by users in a previous study [10], from which we highlight a music composing tool that allows customizing the mapping between movements and audio.

The workflow of our tool comprises of defining rectangles in a videocamera stream from which motion is captured, and then mapping this motion to control customizable sound production agents. This allows to compose motion-based interactions that can be used for audiovisual installations, both virtual and on-site. The whole system executes within a web browser (<https://tiagoft.github.io/motus/>), hence it is not necessary to download and install any additional software in the host computer.

In the future, we will aim at collaborating with potential users of the engine to find new requirements. Also, we will evaluate the impact of using MOTUS and its underlying ideas in several contexts, including art installations, physical therapy, social media, and other possible applications.

References

- [1] Clayton Mamedes. Participação em instalações interativas (in portuguese). *Revista Interdisciplinar Internacional de Artes Visuais*, 5(1), 2018.
- [2] Raul Paiva and Tiago Fernandes Tavares. Impact of algorithmic composition on player immersion in computer games: A case study using markov chains. *Revista Música Hodie*, 18(1):61–73, 2018.
- [3] Todd Winkler. Motion-sensing music: Artistic and technical challenges in two works for dance. In *Proceedings of the International Computer Music Conference*, 01 1998.
- [4] Tiago Fernandes Tavares, Gabriel Rimoldi, Vânia Eger Pontes, and Jônatas Manzoli. Cooperative musical creation using kinect, wiimote, epoc and microphones: a case study with mindsounds. In *Sound and Music Computing 2015*, 2015.
- [5] João Araújo, Avner de Paulo, Igino Silva Junior, Flávio Schiavoni, Mauro César Canito, and Rômulo Costa. A technical approach of the audience participation in the performance 'o chaos das 5'. In *Anais do XVII Simpósio Brasileiro de Computação Musical*, pages 28–34, Porto Alegre, RS, Brasil, 2019. SBC.
- [6] S. Trail, M. Dean, T. F. Tavares, G. Odowichuk, P. Driessen, A. W. Schloss, and G. Tzanetakis. Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect. In *New Interfaces for Musical Expression - NIME 2012*, Ann Arbor, Michigan, U.S.A., May 2012.
- [7] Damián Keller, Cláudio Gomes, and Luzilei Aliel. The handy metaphor: Bimanual, touchless interaction for the internet of musical things. *Journal of New Music Research*, 48(4):385–396, 2019.
- [8] Filipe Calegario, João Tragtenberg, Giordano Cabral, and Geber Ramalho. Batebit controller: Popularizing digital musical instruments' technical development process. In *Anais do XVII Simpósio Brasileiro de Computação Musical*, pages 217–218, Porto Alegre, RS, Brasil, 2019. SBC.
- [9] Cristiano Figueiró, Guilherme Soares, and Bruno Rohde. Esmeril - an interactive audio player and composition system for collaborative experimental music netlabels. In *NIME*, 2019.
- [10] Tiago Fernandes Tavares. An interactive audio-visual installation using ubiquitous hardware and web-based software deployment. *PeerJ Computer Science*, 1:e5, 5 2015.
- [11] Clayton Rosa Mamedes, Denise H. L. Garcia, José Fornari, and Jônatas Manzoli. Abstrações: An audiovisual installation based on motion recognizing. In *2011 4th International Congress on Image and Signal Processing*, volume 2, pages 586–589, 2011.
- [12] Rômulo Vieira and Flávio Schiavoni. Fliperama: An affordable arduino based midi controller. In *Proceedings of Nime 2020*, 2020.
- [13] Cristiano Severo Figueiró. *Desobediência sonora: selos de música experimental e suas tecnologias de sustentabilidade*. EDUFBA, 2019.
- [14] NIME 2020. "installations" accessed jul. 15, 2021. <https://nime2020.bcu.ac.uk/installations/>.
- [15] Mutek. "mutek virtual" accessed jul. 15, 2021. <https://virtual.mutek.org/en/mutek>.
- [16] Ash Turner. "how many smartphones are in the world?" accessed jun. 23, 2021. <https://www.bankmycell.com/blog/how-many-phones-are-in-the-world>.
- [17] zugara.com. "webcam penetration rates adoption" accessed jun. 23, 2021. <http://zugara.com/webcam-penetration-rates-adoption>.
- [18] Shirlene Vianna Moreira and Francis Ricardo dos Reis Justi and Marcos Moreira. Intervenção musical pode melhorar a memória em pacientes com doença de alzheimer? uma revisão sistemática. 2018.
- [19] Thiago Holanda and Paulo José Moté Barboza and Dayana Priscila Maia Mejia. Tratamento fisioterapêutico em pacientes acometidos por alzheimer: Referência bibliográfica. 2012.
- [20] Wyzowl. "why video is exploding on social media in 2021" accessed jul. 14, 2021. <https://www.wyzowl.com/video-social-media-2020/>.
- [21] Zhang Yiming. "homepage" accessed jul. 14, 2021. <https://www.tiktok.com/pt-BR/>.
- [22] JOYO TECHNOLOGY PTE. LTD. "about us" accessed jul. 14, 2021. <https://www.kwai.com/about/>.
- [23] Heather Shipman. Smart art for smart cities. In Miroslav Mateev and Panikkos Poutziouris, editors, *Creative Business and Social Innovations for a Sustainable Future*, pages 251–253, Cham, 2019. Springer International Publishing.
- [24] Processing Foundation. "overview. a short introduction to the processing software and projects from the community." accessed jun. 07, 2021. <https://www.processing.org/overview/>.
- [25] Processing Foundation. "homepage" accessed jun. 07, 2021. <https://www.processingfoundation.org/>.

Design process and rapid prototyping of animated music visualizations

Horhanna Almeida^{1*}, Giordano Cabral² Rute Moura³

¹Mustic[†]- Centro de Informática/Universidade Federal de Pernambuco

Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária(Campus Recife) – 50.740-560, Recife, PE

horhannaa@gmail.com, grec@cin.ufpe.br, rutymaxsuelly@gmail.com

Abstract. *The production of animations for Musical Information Visualization is still scarce and has challenges in the way of visually communicating information. Due to the need for domains of editing software, demanding technical skills and specific knowledge related to each area: animation, music, design and computing. In this article, we present a systematic review of the animated visualization area and, based on its conception processes, we elaborate an experimental model for the creation, prototyping and construction of musical animations. And through sessions developing quick prototypes, where we obtained qualitative results with feedback collection. We conclude that the importance of animation is exceptional, as an ally in the process of designing and creating a musical visualization, as it facilitates the representation of time to communicate structural elements of music, as they are dynamically arranged in a graphic area.*

1. Introduction

This work permeates two major areas: Musical Visualization and Animation. Music Visualization is the area of intersection between Music and Information Visualization, aiming to develop applied studies to communicate musical information with the aid of visual resources [1]. Animation, on the other hand, is a simulation of movements created from the exposure of images, or frames [2].

Specific knowledge alone is not enough to understand the complexity of the studied phenomena. Interdisciplinarity is the combination of knowledge from several different specialties in order to give new approaches to a real problem [3]. This combination of specialties adds value to the process, and it is possible to see that the result obtained by the joint study is more interesting than the sum of the individual contributions of the parts [4].

Given the difficulties in finding research and debates that relate the areas addressed in this research and assuming that animations facilitate the understanding of musical visualizations [1] [5] [6] [7] [8], we verified through a systematic review that several authors recognize the importance of the representation of time through animation in visualizations. As described by Chen, Zhou and Chen [9], information can be represented and arranged as a function of time, in order to understand the chronological order of events or happenings. Heer and Robertson [10] and Robertson et al. [11], reinforce the importance of time representation through animation, as its temporal elements - beginning and end - and movement are highly effective and

easily noticed by peripheral vision, making observers remain oriented during the animation execution, improving interaction and understanding of information.

Although some static visualizations can be visually interesting, to be used in musical analysis they require training, practice and/or sound to be fully understood. For Barisic [12], image, sound or video resources allow the user experience to be more pleasant and interactive, since users have the possibility to discover more information about a certain event.

In an animation, the presence of the temporal element, despite being an ally to fix visual attention and aid in the understanding of information, can be a distraction for the viewer, as the excess of information in motion also hinders the concentration of observers, as well as very slow animations can be tedious Robertson et al. [11]. Therefore, authors such as Baudisch et. al [13]; Tversky, Morrison and Betrancourt [14]; Heer and Robertson [10] still advocate the use of static views, and describe animation as the least effective form of analysis as well as complicating the comparison of flowing items.

Considering this exploratory scenario involving the areas of Musical Visualization and Animation, we realize the importance of using animation in contexts where its presence is not only a creative or entertainment ally, but necessary, as without its presence the understanding of musical information would be misunderstood, as is the case with very popular games: Bandfuse ², Guitar Hero ³, Rock Band ⁴, Rocksmith ⁵ and Audiosurf ⁶. We realized that communicating information through static views is a difficult task [7] and communicating information through animations is an even more difficult task [11] [13] [14] [10], because in addition to the technical difficulties presented by Robertson et al., knowledge in more than one area is necessary, such as: animation, design, music and computing. We also realized that the association of music with animation, improves the user experience and consequently introduces notions about musical information. We emphasize that this joining must happen synchronously, that is, the visual events of the visualization must happen at the same time the music is played and the animation is executed graphically. Thus, this work aims to develop a process of design and rapid prototyping of animated musical visualizations.

²Bandfuse: www.hamster.co.jp/bandfuse/

³Guitar Hero: <https://www.guitarhero.com/de>

⁴Rock Band: www.rockband4.com/

⁵Rocksmith: www.ubisoft.com/en-us/game/rocksmith/pluss

⁶Audiosurf: www.audio-surf.com

*Supported by CNPQ.

[†]<https://mustic.cin.ufpe.br/>

2. Conception prototype design

In order to build animated musical visualizations, we need to immerse ourselves in fast prototyping and conception processes. Rapid Prototyping has as its main resource the ability to quickly build complex physical forms, difficult or even impossible to be built by the technologies established for a long time [15]. Therefore, we will use this method to identify the technical and creative capabilities of developing animated visualizations.

For Canciglieri and Selhorst (2015)[16], it is necessary to understand the limitations of each technology and process used during rapid prototyping, helping in decision making for the design of rapid prototypes, which can happen in three phases:

Phase 1 Descriptions of the product's characteristics: Which are subdivided into two stages, the first of which is an analysis of the physical characteristics of the product with an A/B test to find out which is the best option. And the second, defines the criteria for decision-making in the development of the prototype, with analysis of its use, functionality, available resources, definition of material, dimensions, geometric complexity, total execution time and completion of the prototype. The **Phase 2**, of evaluation of decision-making criteria, evaluates the most relevant characteristics of the product and of the prototyping technologies available for designing the prototype. The composition of these "decision criteria" aims to define the degree of priority for the characteristics presented in Phase 1 of the method, generating a priority ranking through a discrete scale from 0 to 9, with a value of 0 indicating no importance and 9 extreme importance. And finally, **Phase 3**, in which decision-making analysis and final prototyping of the product are performed, taking into account the metrics of the analyzed data and evaluation metrics.

Considering all this, the methodology of this work based on the Design Thinking approach. Essentially defined as a human-centric innovation process that emphasizes observation, collaboration, rapid learning, idea visualization, rapid concept prototyping and simultaneous business analysis, which definitively influences innovation and business strategy [17]. According to IDEO [18], there is no fixed Design thinking process, as the innovation or new idea can go through many iterations before the process is completed. But in summary, considering the many successful efforts include a variation on four intertwined steps: inspiration, synthesis, ideation / experimentation and implementation.

The process of knowing the problem or challenge is inspiration. Synthesis is the process of collecting data and information about the problem or challenge. Ideation/experimentation is the creative process of solving the problem or challenge and implementation/validation is the process of analyzing the prototype.

3. Designing Musical Visualizations

Visualizing music in a meaningful and intuitive way is a challenge Ciuha, Klemenc and Solina [7], as there is a need

for an entire design process that involves a certain technical knowledge about both areas (visualization and music).

The design processes of musical and animated visualizations are used as a basis for systems development and generation of musical visualizations, the musical data inputs can be transmitted via *input* of MIDI files (*Musical Instrument Digital Interface*), as described in the works of [7] [6] [19] [5] [1]. There are also records of graphic experiments created by Design professionals who conceive visualizations based on musical perception, such as HALL⁷ in "Sheet Music Visualization".

Therefore, we identified that there is a graphic choice of visual structure that varies depending on the objective of communicating musical information. For example, as shown in the figure 1 on the left the *Mandrit*, the automatic static visualization generation system for rhythm analysis [1] and on the right the *MusicVis* [19], a web application for automatically generating sound visualization using an MP3 music file or microphone capture as input.



Figure 1: Mandrit end MusicVis

Both use the style Bubble Chart, which are bubble charts to represent musical data by circles in different sizes and colors [20]. Applying graphical form with different goals in static and animated views respectively.

Another step in the process of designing a musical visualization is to understand the best reading orientation. Human beings tend to observe and analyze something visual, from left to right, a pattern naturalized due to writing and reading. For this reason tools bet on this visualization style as well established as HOOK THEORY⁸. Another form of visual reading naturally imposed is clockwise and counterclockwise [21], alluding to the visual representation of the hand clock, existing since the year 1335.

3.1. Conception of Animated Music Views

As well as in the conception of musical visualizations, at this stage the assistance of the areas of Music and Information Visualization is considered. However, the design of animated music visualizations has a third guiding area: animation.

3.1.1. Animation design process

Animation can be defined in different ways. Like a movie made frame by frame, providing an illusion of movement [22]. As a process of dynamically generating a series of

⁷www.behance.net/gallery/6529923/Sheet-Music-Visualization?isa0=1

⁸HOOK THEORY: <https://www.hooktheory.com/>

frames from a set of objects, where each frame is an alteration of the previous frame [23]. For these authors, animation can be a technique in which the illusion of movement is created by photographing a series of individual drawings in successive frames of the film. The illusion is produced by projecting the movie at a certain rate (usually 24 frames/second).

During the progress of animation, Walt Disney and his Disney company had a great prominence, for promoting studies and courses, improving animation techniques and making new artists learn these practices as rules of commerce. After a while, each technique was named and became known as the fundamental principles of animation [24]:

- (1) **Squash and Stretch** - Define the rigidity and mass of an object, distorting its shape during an action.
- (2) **Time** - Spacing actions to set the weight and size of objects and character personality.
- (3) **Anticipation** - The preparation for an action.
- (4) **Staging** - Unambiguously present an idea.
- (5) **Tracking and Overlay Action** - The completion of an action and the establishment of its relationship with the next action.
- (6) **Direct Ahead Action and Pose-to-Pose Action** - The two contrasting approaches to creating movement.
- (7) **Slow In and Out** - The spacing of the intermediate frames to achieve subtlety of time and movement.
- (8) **Arcs** - The visual path of action for natural movement.
- (9) **Exaggeration**- Accentuating the essence of an idea through design and action.
- (10) **Secondary action**- The action of an object resulting from another action,
- (11) **Appeal** - Create a design or action that the public likes to watch.

Knowing what are the principles of an animation and what they are necessary to produce an animation, it is also important to know the processes of designing an animation. Wells [25], in his book "*The Fundamentals of Animation*", says that to be conceived animation must go through 25 stages, these: Concept: the stimulating idea; Creating Work Schedule; Resource review; Research; History; Preparatory View; Formal Design; Storyboard; Road map; Initial soundtrack; Animatic; Filming script; Animation Analysis; Aesthetic Analysis; Layout; Dope Sheet; Development soundtrack; Backgrounds; Animation Sequences; Creation of sequences; Construction; Post-production analysis; Mixing; Output in chosen format and Display.

For Tschang [26], animation has five stages, conceptualization, pre-production, production, post-production and the next look, as shown in the figure 2:

In the conceptualization stage, the main idea and the story are conceived. In the pre-production stage, the conceptual art and the first scenes that will constitute the *storyboard* are developed. In the production stage, it involves most of the intensive work, which includes the art

Stage 1 Conceptualization	Stage 2 Pre-production	Stage 3 Production	Stage 4 Post-production	Next cycle (for tech intensive firms)
Idea for series or production (short concept and final script); planning	Storyboarding; concept art to flesh out script; story reels; script changes	Animating: Very large investment of resources (animators, supporting technical staff), voice acting	Some editing of scenes, retakes	Development of software tools for next generation animated features

Figure 2: Stages of Production Process for General Animation in Tschang

development, modeling and animation itself, as well as the intermediate tasks necessary to obtain it. The stage is post-production where editing and rework are carried out [27].

3.2. Process of designing animated music visualizations

To conceive an animated visualization of a song, all conception processes seen so far must be taken into account: conception of music visualizations and conception of animations.

In the literature, some processes of designing visualizations of synchronous musical animations are reported, such as Musanim⁸ by Stephen Malinowski [6] and others [5] [7] [8].

A relevant example in the study of musical harmony is the Isochords [5], a system for visualizing musical structure in musical animation format that helps in the classification of musical structure. The preview highlights consonant intervals between notes and chords common in music, it also conveys information about interval quality, chord quality, and chord progression synchronously during digital music playback.

Isochords, as shown in the figure 3 uses the representation originally developed by Leonhard Euler - the Tonnetz - to represent the chord progression and communicate visually and musical harmonics, also using the foundation of the circle of fifths as a reference to determine subsets of notes with equivalent intervals, such as pitch signatures, helping to find relative pitches [5].

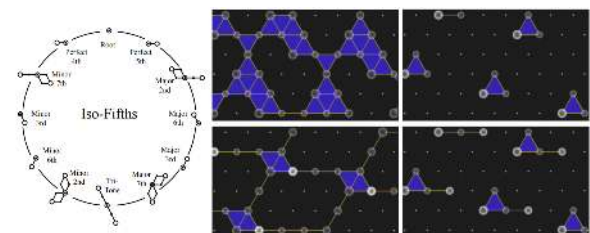


Figure 3: Isochords

The conception of this project takes place through decision making in relation to how the input of information in the system would be done, being defined the MIDI file of the songs with the records of the musical sequences as input data. As for the selection of visual elements, it was defined that the representation of musical information of the graphic choice would be Tonnetz's own structure, using its elements, triangular geometric shapes and color

⁸Musanim: <http://www.musanim.com/Renderers/>

variations. Finally, the animation is performed, inserting the soundtrack and the sequence of the progressions of the Tonnetz triads, using as a reference the time already defined in the beat of the music, conceiving a synchronous musical animated visualization.

Ciuha, Klemenc and Solina [7], produced an animated musical visualization, aiming to interconnect similar aspects in music and visual perception, through the modeling of harmony: affinity of tones and consonance with colors. The authors sought to facilitate the understanding of harmonic relationships in music, which can sometimes be difficult for untrained people to understand, but can be clarified through visual cues 4.

We identified similarities in the process in the conception of animated visualization, as the authors also carry out a study of the graphic choices in relation to the musical information to be represented, namely: the tone scale in a circle of fifths. They extracted information from MIDI files of the songs, attributing in the visual representation the association of musical tones to the colors represented in the circle of fifths 4.

Finally, an animated visualization system in the style piano roll was developed, this is a representation in the timeline type, where the sequences are recorded in a horizontal line. Providing a better understanding of the information that is associated with a period of time, allowing its users to understand what types of events occurred over time [28].

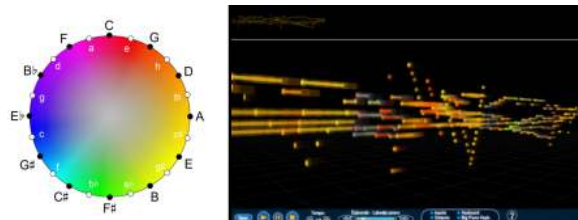


Figure 4: Visualization of concurrent tones in music with colours

Other authors such as Malinowski [6], and Wu et al. [8], also explore the process of designing animated musical visualizations, experimenting, and bringing contributions to the community. Considering this whole exploratory context, we also used as a reference the animation model proposed by Tschang [26], shown in figure 5, Describing the process of creating a musical animation.

Conceptualization	Definition of information
Pre-production	Input definition Definition and production of visual elements (art direction)
Production	Animation realization
Post production	Editing and rework are performed

Figure 5: Process of designing animated music visualizations

Which allows us to structure the process of conception, pre-production, production, and post-production of animation. Thus, standardizing these processes, we can say that to build animated visualizations of music we need to choose the musical information, make production choices such as input definitions, choose the visual elements (art direction), as well as which tools to carry out the animation.

4. Experimental Process

Based on the methodological steps of Design Thinking [18] and the design processes of animated musical visualizations, a process of rapid prototyping of musical animations was developed. In this process, the user will design their own animation, through three phases:

Phase 1 - Inspiration and Synthesis: In this phase, the user will be inserted into the problem, immersing himself in the context and asking the following questions: "If you could see an animated song, which song would you choose and which musical information from the chosen song would you like to portray?". The objective of this phase is to define the choice of music, understand and discuss which musical concepts the user wants to address in his animation.

Phase 2 - Ideation: In this phase the user will answer the questions from the previous phase. When choosing the music, the user will also specify which moment of the music and make sketches of graphical representations creating a prototype of your animation.

Phase 3 - Validation: The authors conceive the animation digitization process and present the graphic result for prototype validation with the expert user who designed the prototype, aiming to analyze whether the final animation meets the proposed objectives considered when performing the prototype.

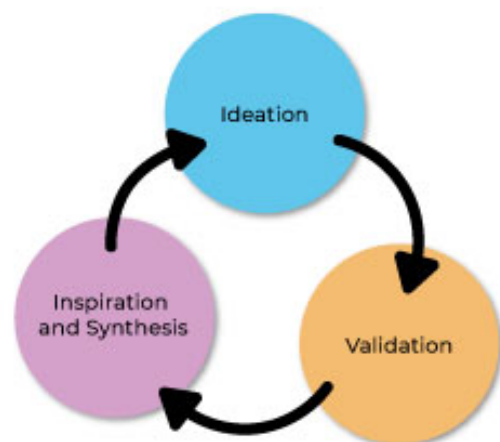


Figure 6: Phases of the Design Thinking methodological process

It is worth noting that in Design Thinking the phases of the methodological process must be repeated as

many times as necessary, revisiting and understanding the already explored content figure 6.

The application of the experimental process will be done through video conference with experts in the field of music. To carry out technical and targeted data collection, aiming to enable new perspectives on how to visualize and communicate musical information through animations. Focusing on potential users with collaborative participation in the experimental process and in the creation of animations.

4.1. Pilot experiment

Here we describe the execution process of the experimental model, with its first session performing a pilot experiment through video conference, with five participants. It is important to note that only one of them had basic knowledge about music, but was not an expert.

Participants took 5 to 10 minutes to choose the music and information they would use in their prototype. At this time, the participants reported difficulty in choosing musical information, as they did not have technical knowledge about music theory. However, all made their choices and the information they mentioned were: BPM, voice, rhythm, tempo, music speed, music intensity, high-pitched sound, bass sound, representation of instruments and voices.

In the prototyping phase, participants used paper and pen to draw their representations. They were instructed to write down all the information on paper, to record such as: song name, color, song lyrics and whatever they thought necessary about the description of the animation. Participants took 10 to 15 minutes to produce their prototypes. The process ended with a shared presentation in which each one explained their animation result.

It is worth noting that all participants were able to understand and perform the experiment, completing it within an estimated time of 30 minutes. We highlight the difficulties recorded by the authors as mediators of the experiment, who, during the individual monitoring of the design process of each participant, faced technical challenges with the video conference camera as it was aimed at people and not at the design, so it was impossible to follow the construction of your sketches.

The pilot experiment served to validate the target audience, realizing that people who are not specialists in music have difficulty in portraying musical concepts. The following experiments were applied only with Music specialists.

4.1.1. Experimental Model: Building Animated Views

After the challenges faced through the pilot experiment, we had technical support to develop a new experimental model that were also carried out by video conference, however with a more controlled digital environment for data collection. Therefore, we requested that the camera be directed to the drawing and we determined an average duration of

10 to 15 minutes. And we held 3 sessions with a music specialist, resulting in 3 experimental prototypes.

In experiment 1, the music specialist followed the methodological phase, choosing the song "Mambembe, by Chico Buarque" seeking to represent the musical information of chord progression. Choosing to cut out your prototype with the first stanza of the song, writing all the chords and assigning through a rectangle the chord pattern that is repeated in pink and the other chords in rectangles in green. The specialist had the idea of creating a visual element that would demarcate the chords for accompaniment, as in the figure 7.

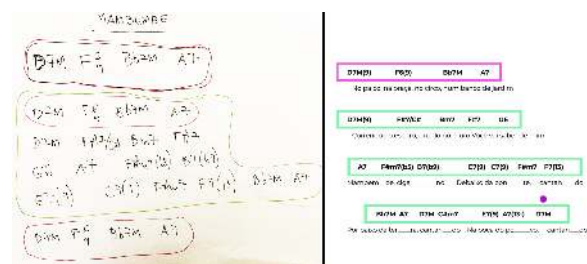


Figure 7: Specialist prototype and animation prototype designed by the authors

Thus, in the validation phase, an animation prototype was generated by the authors according to the information provided by the expert. The first feedback suggested by the expert was to add an indicator on the chords synchronized with the music. As a solution, the pattern was kept throughout the animation (green and pink) and a cursor that jumps in purple was added to make it easier to follow the music.

In experiment 2, as shown in figure 8, the participant chose the song "Piano na Manguera, by Tom Jobim and Chico Buarque". Musical information was the dissonance between harmony and melody.

After automating the animation, the specialist requested a red trail that shows the path taken by the car on the road.



Figure 8: Specialist prototype and animation prototype designed by the authors

While listening to the music, the specialist wrote the part he would like to animate and drew two green paths representing the outline of a road and a red path representing the trail made by a car. The green path was defined as the harmony of the music and the red path the melody of the music. For the participant, the harmony of a song is

sometimes "strange", as it presents a larger set of notes (associated the representation of the road with the more open path) and more obvious (associated the representation of the road with the narrowest path), for the participant, there are moments in the music where the red path (car) leaves the harmony (of the road) and it is particularly interesting to see these moments. The specialist also pointed out that his prototype would be in the timeline style, with the entire road being drawn in a static way and requesting that in the animation only the car would move, however it would have a camera revealing the path and following the movement of the car.

In experiment 3, shown in the figure 9, the specialist chose to analyze the aspect of the music's rhythm: "Brigas Nunca Mais, de João Gilberto".



Figure 9: Specialist prototype and animation prototype designed by the authors

The specialist separated the lyrics of the song into cells, drawing eight diamonds in a figure 9 diamond shape, symbolizing the measures of the music. For the expert, it would be interesting if there was a cursor that represented the notes that run through the cells while the music is playing, that is, when the note is playing, the cursor must be red and the previous note must appear as a "ghost" of reference in yellow color.

After presenting the result of the animation of prototype 3, no adjustments were requested, that is, the participant considered the criterion of prototype fidelity in relation to its sketch.

We conclude with the application of the experimental model, that the possibilities of graphic representation and development of animated musical visualizations are wide. And with rapid prototyping exploring the creative process in a practical way, observing and collecting continuous qualitative feedback. Therefore, we consider that animated visualizations are potential tools to quickly communicate musical information, as it facilitates its real-time and synchronous monitoring in relation to the sound.

5. Discussion

This is a research in progress, but we noticed that the refinement of the process after the application of the pilot experiment directed the new experiments to richer results of information and musical conceptualization. Thus, we highlight some interesting points in the construction of the animations, having musical information as exploratory potentials - the lyrics or the chords of the music -, as it was

frequently present in the temporal reference of the participants' representations. We emphasize that it is necessary to expand, produce and continue this study for the academic community, investigating which are the best prototyping and design techniques, proposing a design pattern for animated music visualizations through rapid prototyping. In future work, we will refine the experimental process and work on the development of an automation software that will reproduce the work of animator that is currently performed by the authors.

References

- [1] Rute Moura. *Visualização Computacional de Informação Musical: Um estudo de caso para análise rítmica*. Dissertação de Mestrado., Universidade Federal de Pernambuco, Recife, 2021.
- [2] Andréia Prieto Gomes. *HISTÓRIA DA ANIMAÇÃO BRASILEIRA*. page 25, 2008.
- [3] Garry D. Brewer. The challenges of interdisciplinarity. *Policy Sciences*, 32(4):327–337, December 1999.
- [4] Homero Dewes. Os desafios da pesquisa e ensino interdisciplinares. *Revista Brasileira de Pós-graduação*, v. 4,:42–65, 07 2007.
- [5] Tony Bergstrom, Karrie Karahalios, and John C. Hart. Isochords: Visualizing structure in music. In *Proceedings of Graphics Interface 2007*, GI '07, page 297–304, New York, NY, USA, 2007. Association for Computing Machinery.
- [6] Stephen Malinowski. Music animation machine renderers, 2012.
- [7] Peter Ciuha, Bojan Klemenc, and Franc Solina. Visualization of concurrent tones in music with colours. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, page 1677–1680, New York, NY, USA, 2010. Association for Computing Machinery.
- [8] Yongmeng Wu, Leshao Zhang, Nick Bryan-Kinns, and Mathieu Barthet. Open symphony: Creative participation for audiences of live music performances. *IEEE MultiMedia*, 24:48–62, 01 2017.
- [9] Bingkun Chen, Hong Zhou, and Xiaojun Chen. E-embed: A time series visualization framework based on earth mover's distance. *Journal of Visual Languages Computing*, 48:110–122, 2018.
- [10] J. Heer and G. Robertson. Animated transitions in statistical data graphics. *IEEE Transactions on Visualization and Computer Graphics*, 13:1240–1247, 2007.
- [11] George Robertson, Roland Fernandez, Danyel Fisher, Bongshin Lee, and John Stasko. Effectiveness of animation in trend visualization. *IEEE Transactions on Visualization and Computer Graphics*, 14(6):1325–1332, 2008.
- [12] Renato Barisic. Cross-curricular education using interactive visualizations. *Procedia - Social and Behavioral Sciences*, 47:2157–2161, 2012. Cyprus International Conference on Educational Research (CY-ICER-2012) North Cyprus, US08-10 February, 2012.
- [13] Patrick Baudisch, Desney Tan, Maxime Collomb, Dan Robbins, Ken Hinckley, Maneesh Agrawala, Shengdong Zhao, and Gonzalo Ramos. Phosphor: Explaining transitions in the user interface using afterglow effects. In *Proceedings of the 19th Annual ACM Symposium on User Interface Software and Technology*, UIST '06, page 169–178, New York, NY, USA, 2006. Association for Computing Machinery.

- [14] Barbara Tversky, Julie Bauer Morrison, and Mireille Be-trancourt. Animation: Can it facilitate? *Int. J. Hum.-Comput. Stud.*, 57(4):247–262, October 2002.
- [15] Marcelo Fernandes de Oliveira. *Aplicações da prototipagem rápida em projetos de pesquisa*. Mestre em Engenharia Mecânica, Universidade Estadual de Campinas, Campinas, December 2008.
- [16] Osiris Canciglieri Junior, Aguilar Selhorst Junior, and Ângelo Márcio Oliveira Sant’Anna. Método de decisão dos processos de prototipagem rápida na concepção de novos produtos. *Gestão & Produção*, 22(2):345–355, June 2015.
- [17] Thomas Lockwood. *Design thinking : integrating innovation, customer experience and brand value*. New York, NY : Allworth Press [u.a.], 3 edition, 2010.
- [18] IDEO: Design Thinking, 2018.
- [19] Van Huynh and David Comberg. Musicvis a web application for visualizing sound., 2017.
- [20] Leandro Soares Guedes. *MusicVis: Interactive Visualization Tool For Exploring Music Rankings*. Dissertação de Mestrado., Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- [21] LANDR. O Círculo de Quintas: O que é e como usá-lo em suas composições, October 2020.
- [22] Jill Neldes. *Introduction to Film Studies*. London: Routledge, 5 edition, 2011.
- [23] N.Magnenat Thalmann and D. Thalmann. Computer graphics and animation. *Computer Physics Reports*, 11(1):221–291, 1989.
- [24] John Lasseter. Principles of traditional animation applied to 3d computer animation. *SIGGRAPH Comput. Graph.*, 21(4):35–44, August 1987.
- [25] P. Wells. *The Fundamentals of Animation*. Fundamentals (Ava) Series. Bloomsbury Academic, 2006.
- [26] T. Tschang and A. Goldstein. Production and political economy in the animation industry: Why insourcing and outsourcing occur. 2004.
- [27] Elisangela Lobo Schirigatti and Ilan Avrichir. Processo de produção de uma animação: Uma abordagem teórica. *Diálogo com a Economia Criativa*, 4(11):32, August 2019.
- [28] Richard Lipka. Scalable timeline visualization. In *2016 9th International Conference on Human System Interactions (HSI)*, pages 150–156, Portsmouth, United Kingdom, July 2016. IEEE.

Electric guitar distortion effects unit using a Raspberry Pi

Renato Santos Pereira¹, Rodrigo Varejão Andreão²

¹Centro de Pós-Graduação – Centro Universitário Espírito-Santense (FAESA)
Av. Vitória, 2220 – 29053-360 Vitória, ES

²Departamento de Engenharia Elétrica – Instituto Federal do Espírito Santo
Av. Vitória, 1729 – 29040-780 Vitória, ES

renato.stosp@gmail.com, rodrigova@ifes.edu.br

Abstract. *With the advance of electronics, techniques and algorithms for digital signal processing, digital equipment has been gaining more and more space in the music scene. Micro-processed tools now generate several effects such as modulation, echo, and distortion of sounds generated by musical instruments, previously obtained only by analog units. In this context, this study aimed to develop a prototype of distortion effects unit using a Raspberry Pi (a low-cost small single-board computer) and affordable electronic components. Therefore, five nonlinear functions were used, four of which are present in the literature and one of them was originally developed by the authors. These functions model the behavior of an active element (such as transistors, valves, and operational amplifiers), which when they exceed their amplification thresholds produce distortions in the audio signals. Throughout this article, all the steps in the development of the analog circuits for signal acquisition and output will be presented, as well as the simulation and implementation of the functions in the microcontroller. At the end, with the finished prototype, the frequency response analysis is performed and the sound results achieved by the algorithms is compared with each other and with other distortion units.*

1. Introduction

Until the emergence of modern science, musical instruments and theaters were built based only on empirical knowledge [1]. Scientific studies in this area, over time, directly influenced musical production. And from the intersection between science and music, the guitar emerged.

The electric guitar had its origin in the 1930s. The first solid-body guitar model, known as “Frying Pan”, was developed by Adolph Rickenbacker and George Beauchamp, the latter being the inventor of the magnetic pickup [2]. Distortion, one of the most used effects by guitarists, originated with tube amplifiers. When they became popular, amplifiers for electric instruments used thermionic valves in their circuits.

When tube amplifiers operate beyond their maximum amplification capacity (nonlinear operation) they promote signal distortion. The characteristic distortion of this equipment became attractive to musicians in the 1960s and 1970s because it reinforced the harmonic content of chords and notes in a peculiar way [3]. With the emergence of new musical styles from the late 1960s on, additional distortion became a necessity for guitarists. Thus, distortion pedals were developed - electronic equipment used to

add more distortion to the electric guitar signal [3]. With the advancement of digital signal processing techniques, starting in the 1990s, techniques for simulating amplifiers and distortion units were used.

2. Nonlinear Distortions

Physical systems can be mathematically modeled as linear or nonlinear systems [3]. Linear systems are those that satisfy the Superposition Principle [4]. In addition, they do not change the frequency response characteristics of their inputs, in other words, they do not add harmonics to the original signal.

The distortion effect is a widely used effect in the musical world, which is an example of a nonlinear system. Although its most common use is in guitar effects, it is also commonly applied in electric bass guitars, keyboards and even vocals [5]. One way to obtain the distortion effect is by increasing the gain of the input signal, which causes the saturation of the amplifier element. Such a process is called clipping [5]. Another possibility to generate the distortion effect of an audio signal can be modeled mathematically, as a time series generated by some nonlinear function [3], which adds odd or even harmonic components, depending on the type of distortion. This form is known as waveshaping [5].

Waveshaping functions are usually static and although it does not exactly match the physical characteristics of real circuits, this approach can be considered satisfactory and is used in commercial amplifier simulators. The use of static nonlinear functions has the purpose of reducing computational effort and, consequently, making real-time simulation feasible [3].

We can classify distortion effects, in relation to the symmetry of the output waveform, as symmetric or asymmetric. Symmetric distortion occurs when the positive and negative parts of the audio signal are limited at the same amplitude, while asymmetric distortion occurs when the positive and negative parts are limited at different amplitudes. This determines whether the generated harmonics will be only odd (symmetric clipping) or odd and even (asymmetric clipping) [6]. According to the clipping, the effect can be classified as soft clipping or hard clipping. In soft clipping, the output signal starts to be soft clipped as it approaches an amplification threshold. In hard clipping, the output signal is clipped just after it exceeds a certain threshold [7]. This determines whether most of the resulting harmonics are high-order or low-order harmonics [6].

2.1. Distortion Functions

There is a wide variety of distortion function formats, symmetric and asymmetric, smooth and discontinuous functions [1], of which five were chosen for study, simulation and implementation in the microcontroller. The first distortion function, presented in Equation 1 is a soft clipper based on the hyperbolic tangent [8]:

$$f(x) = \tanh x \quad (1)$$

A second distortion function is a soft clipper based on the arc-tangent as shown in Equation 2 [1]. In this model, k_d represents the distortion coefficient. Such a coefficient is directly proportional to the amount of distortion applied.

$$f(x) = \frac{\tan^{-1}(k_d x)}{k_d \pi} \quad (2)$$

A third distortion function is the hard clipper, presented in Equation 3 [9]. In this model, the parameter a represents the value at which the input signal is clipped.

$$f(x) = \begin{cases} x, & |x| \leq a \\ a, & \text{otherwise} \end{cases} \quad (3)$$

A fourth distortion function is the cubic soft clipper, described by Equation 4 [10].

$$f(x) = \begin{cases} -\frac{2}{3}, & x < -1 \\ x - \frac{x^3}{3}, & -1 \leq x \leq 1 \\ \frac{2}{3}, & x > 1 \end{cases} \quad (4)$$

The author in [11] cited the asymmetric clipping equation, described by Equation 5, as proposed to simulate triode distortions. In this model, $dist$ is a distortion tuning parameter and Q simulates the operating point of the triode.

$$f(x) = \frac{x - Q}{1 - e^{-dist(x-Q)}} + \frac{Q}{1 - e^{-dist(Q)}}, \quad (5)$$

$Q \neq 0, x \neq Q$

The design parameters for the asymmetric distortion simulation are based on the above-mentioned function, in which no distortion should occur when the input level is low (the derivative of $f(x)$ should be $f'(0) = 1$ and $f(0) = 0$).

With a limited sampling rate, a high gain on a sine input can produce an approximately square wave output, regardless of the distortion function. All saturating nonlinearities approach a hard clipper asymptotically as the gain increases. There are spectral differences between these different distortion functions, which are subtle but noticeable at lower gain levels. For high gain levels, the soft clipping and hard clipping functions are very similar in terms of output spectrum. The arc-tangent function also exhibits high saturation at large input levels, but the smooth transition produces a smoother spectral response. Hyperbolic

tangent functions saturate more slowly at large input levels, thus reducing high-order distortion and aliasing compared to the arc-tangent function [8]. The asymmetric distortion function must perform clipping and limiting large negative input values, while for positive values the behavior is approximately linear. This equation produces even and odd order harmonics [11].

3. Distortion Functions Simulation

In this step, the five equations presented in Section 2.1 of this paper was simulated by software. Three of them (the soft clipper based in hyperbolic tangent, the cubic soft clipper and the asymmetric clipping function) have been adapted to make the project more functional when implemented in the microcontroller. In the other two functions (the soft clipper based on the arc-tangent and the hard clipper), no adjustment was needed. In order that, in the soft clipper based on the hyperbolic tangent, detailed in Equation 1, a coefficient k was added to modify the amount of distortion. This, when multiplied at the input x , modifies the amount of distortion produced by this function. Therefore, the new function based on this equation is described by Equation 6:

$$f(x) = \tanh kx \quad (6)$$

The soft clipper based on the arc-tangent, detailed in Equation 2, has a coefficient k_d , which controls the amount of distortion applied to the input signal, in a directly proportional manner. In the hard-clipping function, detailed in Equation 3, the coefficient a controls the amount of distortion applied to the input signal. The smaller the modulus of a , the more distortion is obtained. From the cubic soft clipper, detailed in Equation 4, a new polynomial function was developed, with a coefficient k , which controls the amount of distortion applied, in an inversely proportional manner, as follows:

$$f(x) = x - \frac{x^k}{k} \operatorname{sgn} x, \quad k \in \mathbb{N}, -1 \leq x \leq 1 \quad (7)$$

The signal function was inserted into the system because for odd values of k :

$$\begin{cases} \frac{x^k}{k} < 0, & x < 0 \\ \frac{x^k}{k} \geq 0, & \text{otherwise} \end{cases}$$

However, for even values of k , $\frac{x^k}{k} \geq 0$, even for $x < 0$. In this way, the original function, detailed in Equation 4, would not produce the desired effect at the input. Finally, for the asymmetric clipping function, detailed in Equation 5, a new function (Equation 8) was developed, also with a coefficient k , that controls the amount of distortion applied.

$$f(x) = 0,5 + x - e^{-kx} \frac{1}{|\min x - e^{-kx}|} \quad (8)$$

The software used for the analysis of each of the models was MATLAB®. The input signal was a sinusoid,

with unity amplitude and frequency of 110 Hz. The sampling rate used was 8 kHz, the default of the program. In addition to the output waveforms, the frequency response of each function was also generated as a function of the input signal. The method used to obtain the frequency response was the Fast Fourier Transform (FFT). It is also worth noting that after passing through the system, the signal receives a gain, so that its amplitude is unitary. The amplitude and frequency charts of the five functions is shown in Figures 1 to 5.

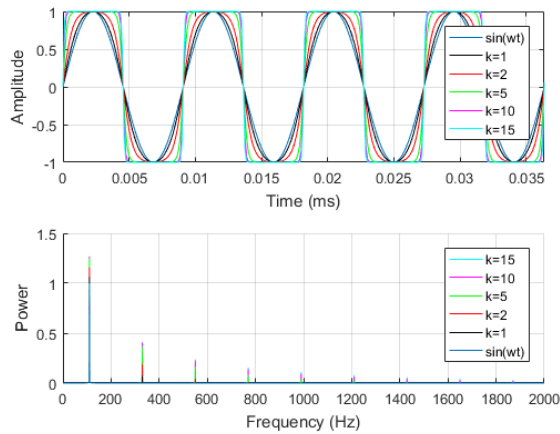


Figure 1: Hyperbolic tangent - Time and frequency domain charts

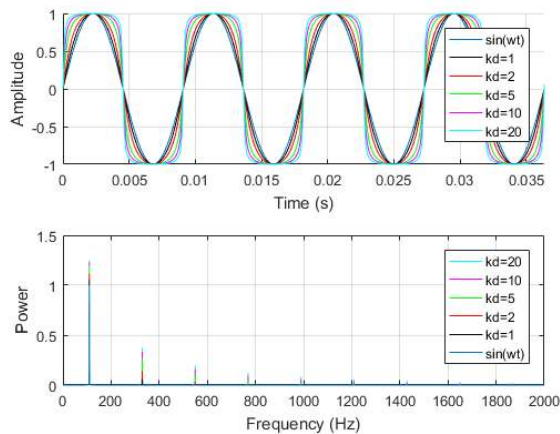


Figure 2: Arc-tangent - Time and frequency domain charts

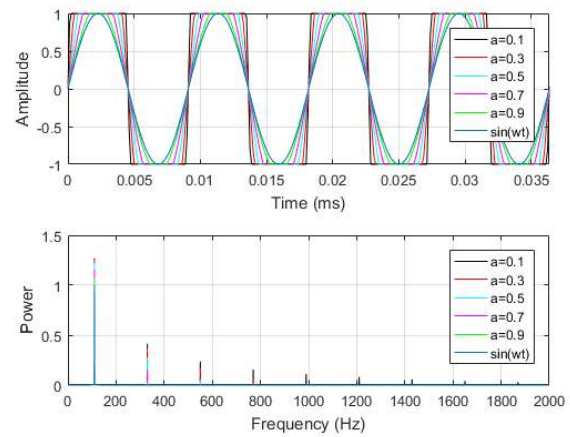


Figure 3: Hard clipper - Time and frequency domain charts

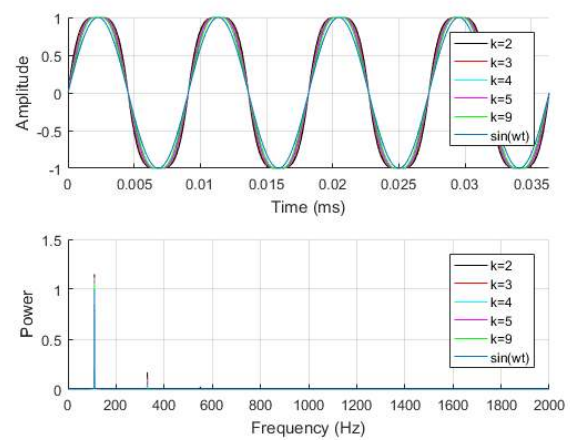


Figure 4: Polynomial Soft Clipper - Time and frequency domain charts

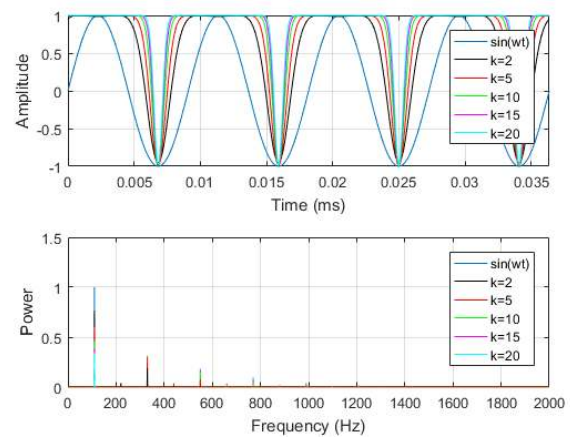


Figure 5: Asymmetric clipping - Time and frequency domain charts

Note that in the models characterized by symmetrical clipping (hyperbolic tangent - Figure 1[12], arc-tangent - Figure 2[12], hard clipper - Figure 3[12] and soft clipper - Figure 4[12]), there is the presence of only the odd harmonics, especially the third and fifth harmonics. Furthermore, it can be observed that the increase in distortion applied to the signal implies an increase in power of the higher harmonics. As the function based on the cu-

bic soft clipper applies a softer distortion, the presence of these harmonics becomes less evident. It can also be seen that, except for the asymmetric distortion function (Figure 5[12]), the output signals of the system have higher power than the power of the input signal. In the asymmetric distortion function, both odd and even harmonics are present. The power of the fundamental harmonic decreases and this power is distributed among the other harmonics.

4. Analog Circuit Development

In this step, the input stage circuit (responsible for acquiring the guitar signal to be processed by the Raspberry Pi) and the output stage circuit (in which the already processed signal is prepared for be received by another effect unit, an amplifier or a soundboard) was simulated and mounted on a breadboard (Figure 6). These circuits were built based on the Pedal Pi project, developed by ElectroSmash [13]. This project was chosen as a basis for the relative simplicity of the circuit, as well as the ease of obtaining the electronic components used in the prototype. The software used for the circuit simulation was the Multisim®.

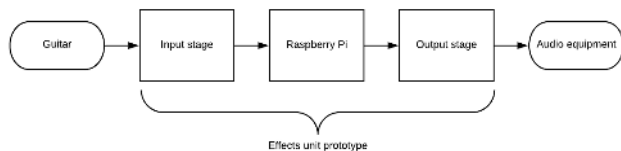


Figure 6: Distortion effects unit - block diagram

4.1. Input Stage Development

The signal acquisition circuit (Figure 7[12]) consists of an operational amplifier, in the non-inverting configuration, followed by a bandpass filter. The signal, which initially has a voltage level between 100 mV and 2 V, receives a gain (adjustable between 1 and 20 by a 500 k Ω potentiometer) so that it can be optimized for the analog-to-digital converter (ADC). In addition, this part of the prototype is responsible for shifting the offset of the signal, so that the SPI converter and consequently the Raspberry Pi do not receive negative voltage values, which would damage the device. The bandpass filter is composed by two first-order low-pass filters and two first-order high-pass filters. The two first-order low-pass filters function is to attenuate higher frequencies, which can generate aliasing. The two first-order high-pass filters are responsible for removing the mains voltage noise and the DC component of the signal.

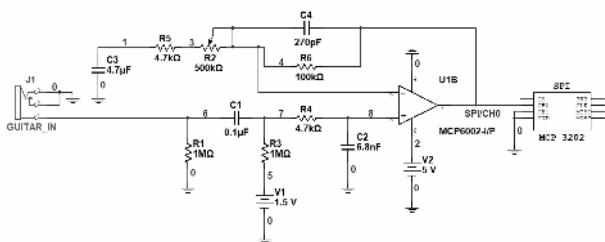


Figure 7: Signal acquisition circuit

To simulate the frequency response (Figure 8[12]) and the amplitude graph (Figure 9[12]) of this circuit, the

input signal was modeled as a sinusoid, with a voltage of 2 V_{pp} (0.707 VRMS) and a frequency of 440 Hz. Potentiometer R2 was set so that the voltage gain of the operational amplifier had its minimum value. This resulted in a gain of approximately 1.12, or approximately 1 dB. This causes the curve corresponding to the system output to be shifted upwards by 1 dB. Therefore, one can consider as cutoff frequencies of the bandpass filter those whose attenuation is at -2 dB. These frequencies, ω_1 and ω_2 are approximately 1.5 Hz and 4.65 kHz. According to the amplitude plot, it can be seen that the signal at the output of the amp-op has a voltage of 2.25 V_{pp} an offset of approximately 1.6 V.

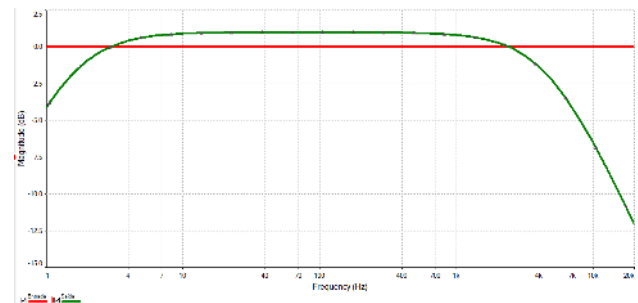


Figure 8: Signal acquisition circuit - Frequency response

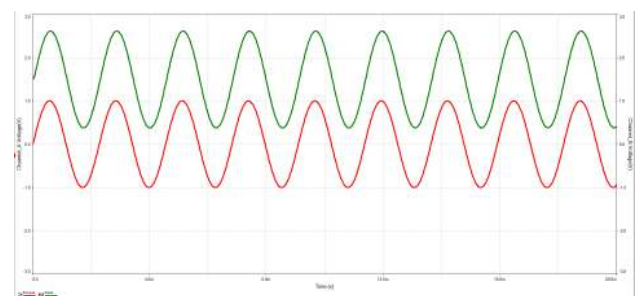


Figure 9: Signal acquisition circuit - Amplitude graph

After simulating this stage, the circuit was mounted on a breadboard and connected to an oscilloscope. A sinusoid generated by the oscilloscope was used as the input signal, with a voltage of 2 V_{pp} and frequency of 440 Hz. By adjusting the potentiometer to obtain the minimum gain value, an output voltage of approximately 2.3 V_{pp} was obtained. As it was observed in the simulation, the output signal of this stage has an offset of approximately 1.6 V. With the potentiometer set to obtain the maximum gain value, the voltage of the input signal reduced to 160 mV_{pp}. The measured voltage at the output of the acquisition stage was 3.2 V_{pp}. Since the Raspberry Pi Zero W does not read analog quantities, after the conditioning stage the signal goes through an external analog-to-digital converter, to then be received by the board. For this step, the MCP3202 was used, an ADC with 12-bit resolution and SPI interface, from Microchip. The reference voltage adopted in this element was 3.3 V, as this is the maximum voltage supported by the Raspberry Pi in its I/O pins. As the MCP3202 has limiting diodes in its internal circuitry, the use of diodes for microcontroller protection

is not necessary.

4.2. Output Stage Development

After the signal passes through the acquisition stage and the microcontroller applies the desired distortion effect, the signal generated by the Raspberry Pi's PWM outputs passes through the output stage. This part of the prototype consists of a third-order active filter (of Sallen-Key topology, shown in Figure 10[12]), a tone control circuit and an operational amplifier in non-inverting configuration with variable gain. To illustrate the frequency response of the Sallen-Key filter, magnitude and phase plots were generated in MATLAB® software. According to these, shown in Figure 11[12], the cutoff frequency is approximately 4.9 kHz. This results in the removal of high-order harmonics, which can generate noise in the desired signal.

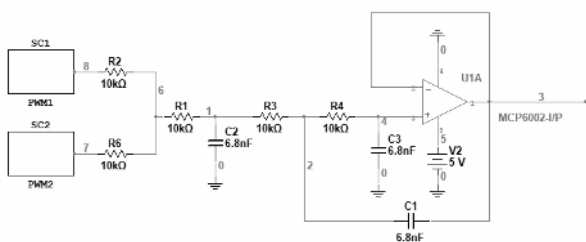


Figure 10: Sallen-Key filter circuit

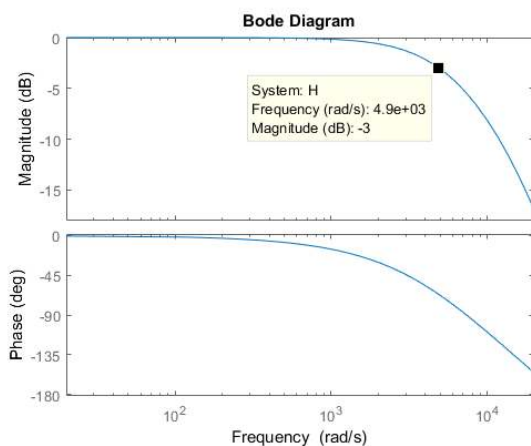


Figure 11: Sallen-Key filter - frequency response

The tone adjustment circuit is composed of the parallel association between a low-pass filter and a high-pass filter, where potentiometers select the bandwidth of the two filters. One feature that was observed during the simulation of this part of the circuit is that there is an attenuation of the signal throughout the selected bandwidth (from 20 to 20 kHz), especially in the region of mid frequencies (around 1 kHz), regardless of the setting of the potentiometers. This is due to the sum of the responses of the two filters used in the tone adjustment. To correct this effect, an operational amplifier was inserted in series with the filters, in the non-inverting configuration, with a 10 kΩ potentiometer for gain adjustment. The schematic of this circuit is shown in Figure 12[12]. The frequency response of the tone control circuit, shown in Figure 13[12],

was generated in the software Multisim®, with the adjustment potentiometers of the two filters adjusted to expand the signal passband as much as possible. The potentiometer connected to the operational amplifier was adjusted so as to obtain a gain of 12 dB. The curve in red represents the signal at the input of the operational amplifier, while the curve in blue represents the signal at the output of the amp-op. With this stage simulation completed, the circuit was mounted on a breadboard.

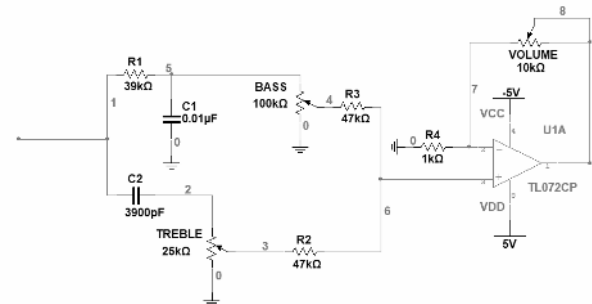


Figure 12: Tone and volume adjustment circuit

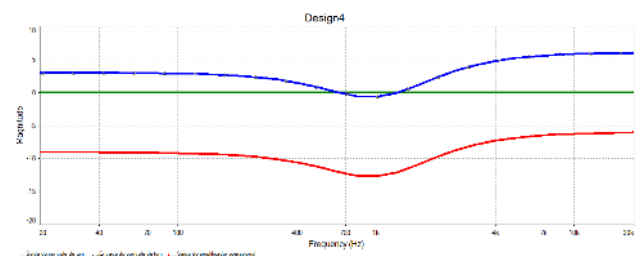


Figure 13: Tone adjustment circuit - Frequency response

5. Raspberry Pi Implementation

Parallel to the study of each system, its characteristics and its implementation computational feasibility, different prototyping platforms were analyzed (Arduino® DUE Raspberry Pi, BeagleBoard, etc.) and it was defined that the hardware used would be a Raspberry Pi, Zero W model. The choice of this hardware was motivated by its low cost, easy access, processing capacity and memory (it has a single-core ARM BCM2835 processor with 1GHz clock and 512 MB RAM), considered sufficient for this study. The Raspberry Pi Zero W has 40 programmable input and output pins, called General Purpose Input/Output (GPIO). The connections of this equipment to the input and output circuits are made through these pins. The operating system used on the board was Raspbian Stretch Lite, a free software based on the Debian Stretch OS that has only the command line interface (or Terminal). After the simulation and breadboard assembly of the analog input and output circuits, the algorithms implemented on the Raspberry Pi relative to each of the five distortion models were developed. For this step, it was necessary to install the bcm2835 library, a C programming language library that provides access to the GPIO pins, allowing the reading and configuration of digital inputs and outputs, serial communications via SPI and I2C interfaces, and access to the system timers. As the programming language adopted in this stage of the

project was the Python language, it was the Python 3 programming environment had to be installed, as well as the PyBCM2835 (Python extension of the bcm2835 library) and NumPy libraries. In addition, the Git version control system was added to the operating system. It is worth noting the lack of documentation for the PyBCM2835 library, which made this task difficult. The algorithm that served as the basis for the elaboration of the codes for each distortion effect, present in the Pedal Pi project, was developed in the C programming language. That said the strategy adopted was to transcribe the base code to the Python language and modify the transcribed code so that it could implement each effect in an optimized way. The compatibility of the data types from the C language to the Python language was accomplished by means of the ctypes library. The sequence of instructions of the codes implemented in the microcontroller can be seen in the flowchart shown in Figure 14.

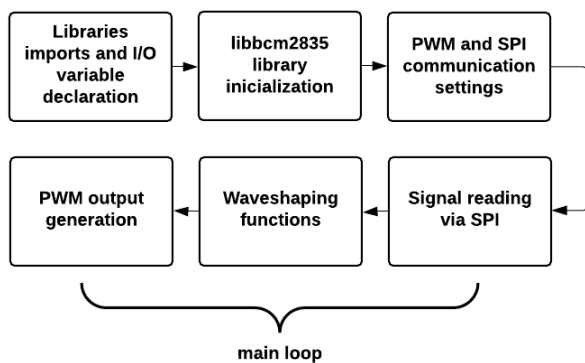


Figure 14: Flow chart of distortion algorithms

In this project, the reconstruction of the signal after applying the mathematical models is done by means of two PWM signals in parallel. One of the PWM outputs is responsible for the D/A conversion of the 6 most significant bits, while the other output converts the 6 less significant bits. The sampling rate, bounded by the supply voltage of the MCP 3202, is approximately 50 kHz [14]. The clock used for SPI communication was 2 MHz and the switching frequency of the PWM signals used was 150 kHz. Subsequently, the five distortion algorithms were hosted via Git in a private repository on the GitHub platform, so that the author could make later modifications remotely.

6. Results

The distortion effects unit was tested through the following experiment: the prototype was assembled according to the block diagram shown in Figure 6. Initially, the input signal was modeled as a sinusoid, generated by an oscilloscope, with a voltage of 2Vpp (0.707 VRMS) and a frequency of 440 Hz. It was observed that the interference caused by the AC mains hum was very significant, which made it impossible to visualize the harmonics added by the distortion algorithms. Different filtering techniques were adopted in

an attempt to attenuate that interference, but these did not have the desired effect. The solution adopted to obtain a better visualization was to change the input signal frequency to 120 Hz.

It can be noticed that, among the symmetric distortion models, the presence of the odd harmonics is more evident in the hard clipping function (Figure 19[12] and Figure 20[12]), in which it is possible to clearly see the third, fifth and seventh harmonics. In contrast to this function, the soft clipper (Figure 21[12] and Figure 22[12]) was the function in which the upper harmonics were least evident. In the other two models (hyperbolic tangent - Figure 15[12] and Figure 16[12] and arc-tangent - Figure 17[12] and Figure 18[12]), it is possible to see the third harmonic more clearly.

In the asymmetric distortion function (Figure 23[12] and Figure 24[12]), it is possible to notice the presence of both odd and even harmonics. This demonstrates that the simulation results and the results obtained in the tests with the finalized prototype are convergent.

Finally, with the finished prototype, tests were performed with an electric guitar in order to evaluate the sound created by each effect, as well as its characteristics in relation to the frequencies of notes and chords of the instrument. It was possible to notice the presence of AC mains hum, also influenced by the single-coil pickup present in the guitar used. The audio sample of each effect are available on [15].

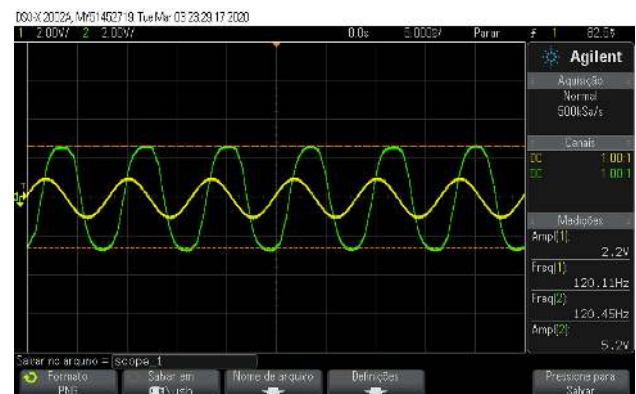


Figure 15: Hyperbolic tangent - Amplitude graph ($k = 2$)

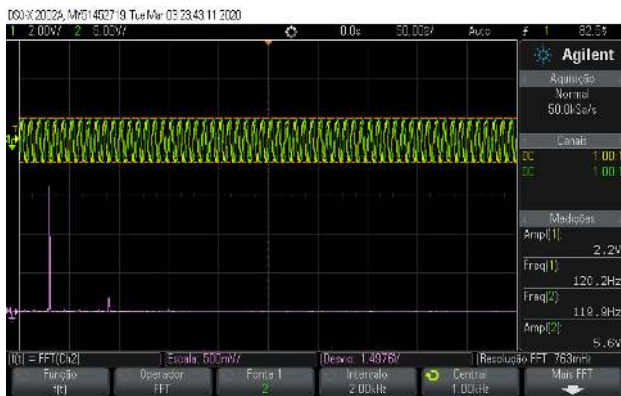


Figure 16: Hyperbolic tangent - Frequency response ($k = 2$)

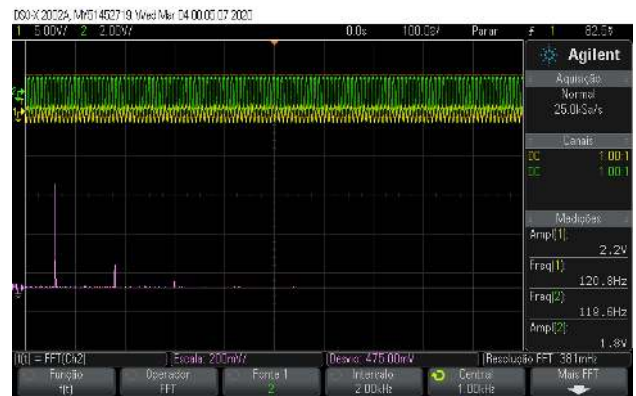


Figure 20: Hard clipping - Frequency response ($a = 500$)

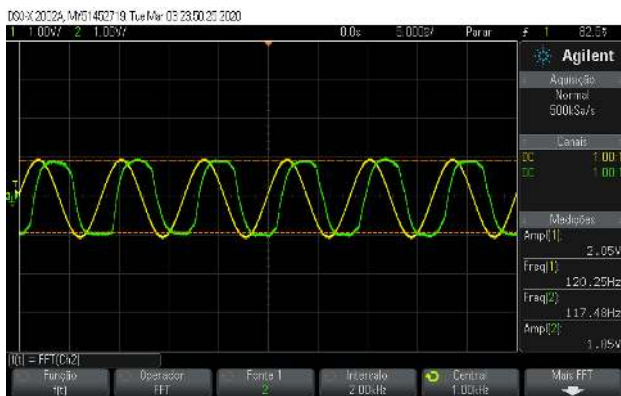


Figure 17: Arc-tangent - Amplitude graph ($k_d = 5$)

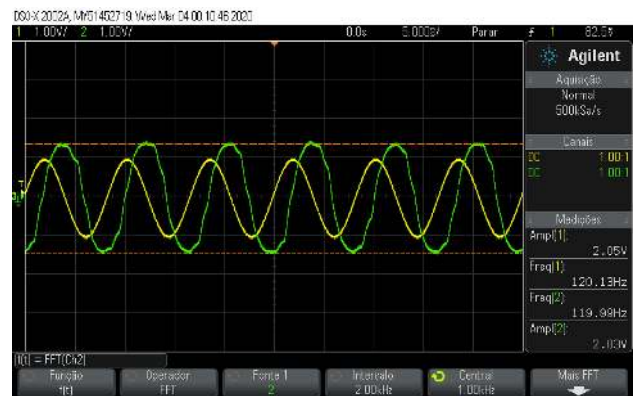


Figure 21: Soft clipping - Amplitude graph ($k = 2$)

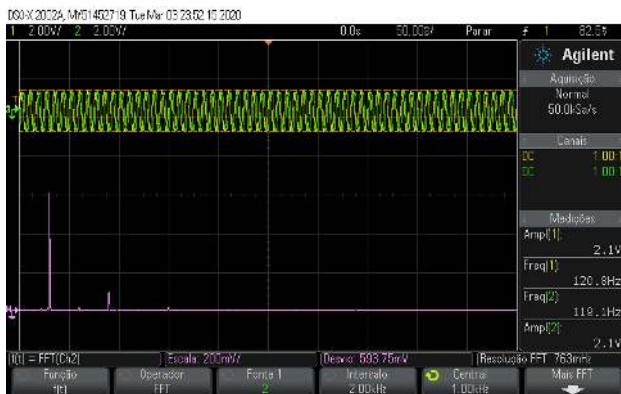


Figure 18: Arc-tangent - Frequency response ($k_d = 5$)

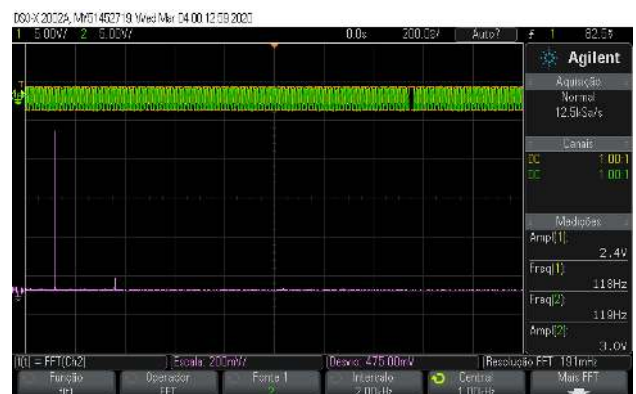


Figure 22: Soft clipping - Frequency response ($k = 2$)

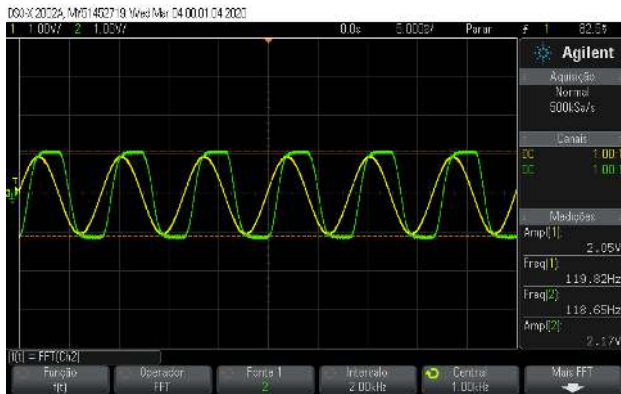


Figure 19: Hard clipping - Amplitude graph ($a = 500$)

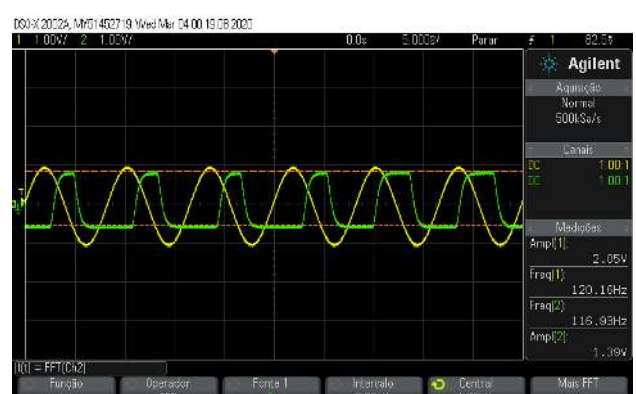


Figure 23: Asymmetric clipping - Amplitude graph ($k = 5$)

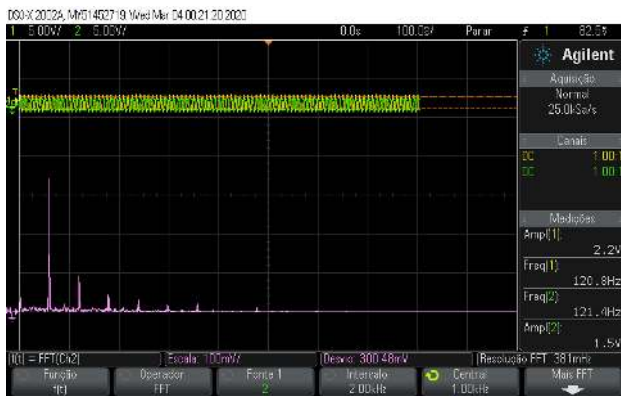


Figure 24: Asymmetric clipping - Frequency response ($k = 5$)

7. Conclusion

The purpose of this work was to design a microprocessor-based distortion effects unit for electric guitar. Through the realization of each step proposed in this study, the overall goal was completed, which can be seen by analyzing the results achieved at each stage of development. It was possible to see the convergence of the theory presented with the results obtained in practice. From the author's point of view, the sound results achieved by each of the algorithms were quite similar to each other. Even though the performance of the prototype was within expectations, it is possible to realize that the sound result obtained through the unit designed in this work falls short of the audio generated by analog equipment or digital equipment of higher value.

It is important to note that professional equipment with the same purpose of this prototype - sound mixers, pedals and digital amplifiers - use DSP's with their own specifications for such activity. In addition, the embedded software used in these units have complex techniques for simulating analog effects and noise elimination. Finally, such devices have electronic components of higher quality than those used in this project.

References

- [1] Nicolau Leal Werneck. *Análise da distorção musical de guitarras elétricas*. Dissertação (mestrado), Universidade Estadual de Campinas - Faculdade de Engenharia Elétrica e de Computação, 2007.
- [2] Neigmar de Souza. Guitarra elétrica: um ícone na cultura pop do Século XX. *Revista Vernáculo*, 2002.
- [3] Thomaz Chaves A. Oliveira, Gilmar Barreto, and Alexander Mattioli Pasqual. Modelagem computacional de efeitos de distorções não lineares para guitarra elétrica. *Revista Brasileira de Computação Aplicada*, pages 69–84, 2013.
- [4] John G Proakis. *Digital Signal Processing: Principles Algorithms and Applications*. Pearson Education India, 2007.
- [5] André Wagner França. *Uso de processamento digital de áudio na implementação de efeitos em instrumentos musicais*. Monografia (Bacharelado em Ciência da Computação), Universidade de Brasília, 2015.

- [6] Robert A Moog. Distortion sound effects circuit, December 25 1979. US Patent 4,180,707.
- [7] Pedro Miguel Cruz and Nuno Borges Carvalho. A comprehensive analysis of the clipping effects on signals with different statistical patterns. In *2014 International Workshop on Integrated Nonlinear Microwave and Millimetre-wave Circuits (INMMiC)*, pages 1–3. IEEE, 2014.
- [8] David Te-Mao Yeh. *Digital implementation of musical distortion circuits by analysis and simulation*. Ph.D thesis), Stanford University, 2009.
- [9] Michel Doidic, Michael Mecca, Marcus Ryle, Curtis Senffner, et al. Tube modeling programmable digital guitar amplification system, August 4 1998. US Patent 5,789,689.
- [10] Julius Orion Smith. *Physical audio signal processing: For virtual musical instruments and audio effects*. W3K publishing, 2006.
- [11] Pierre Dutilleul and Udo Zölzer. Nonlinear processing. *DAFX: Digital Audio Effects*, 2004.
- [12] Renato Santos Pereira. *Projeto de uma unidade de efeitos de distorção para guitarra elétrica*. TCC (graduação em Engenharia Elétrica), Instituto Federal de Educação, Ciência e Tecnologia do Espírito Santo, 2020.
- [13] Andrew Gregory. Review: ElectroSmash Pedal Pi — HackSpace Magazine, 2018.
- [14] MICROCHIP TECHNOLOGY INC. 2.7V Dual Channel 12-Bit A/D Converter with SPI Serial Interface, 2006.
- [15] Renato Santos Pereira. Audio samples: Electric guitar distortion effects unit using a Raspberry Pi [Album]. <https://soundcloud.com/renatostosp/sets/prototype-audio-samples>, 2019.

Challenges to generate musical visualizations

Rute Moura¹*, Giordano Cabral¹, Jader Abreu¹, Mychelline Cunha¹, Horhanna Almeida¹

¹Music[†] Centro de Informática/Universidade Federal de Pernambuco

Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária(Campus Recife) – 50.740-560, Recife, PE

rutymaxsually@gmail.com, grec@cin.ufpe.br, jaoa@cin.ufpe.br, msh@cin.ufpe.br, hao@cin.ufpe.br

Abstract. *The technological acceleration of recent years has allowed for significant advances in data processing and the study of Music, an art with richly expressive and communicative information. In this article, we bring reports of experiences and contributions in the treatment of musical data extracted from digital MIDI and sound files, for the development of systems that generate musical visualizations.*

1. Introduction

Music enchants and fascinates humanity for a long time. It is capable of arousing feelings and emotions, it contains abstract data of subjective characteristics that have an innate complexity to be measured and analyzed.[1] The explanations of its concepts in Musicology literature are based on dense and long technical texts of music theory with few explorations of visual resources, deepening the study of musical information as an object of communication, and using very few computational resources available to facilitate the apprehension, perception, and analysis of its elementary concepts.

Given the computational capacity in data processing today, the ease of access to open data increased the power of the music industry through Peer-to-peer (P2P) sharing and access to collections of musical works through digital files such as MP3, WAV, MIDI. However, due to the large volume of existing compressed raw data, there are often computational complexities in the treatment and extraction of characteristics from musical information to make them visually communicable.

Within this context and considering the representational challenges of Music, members of the research group Music-CIn/UFPE, seek research projects to assist in the visual communication of musical information. Using software resources generating useful applications, enabling technology to be a means to help professionals and experimenters in the area of Music in their creative processes and to expand access to musical information.

2. Visual Music

Computational Music Visualization is an interdisciplinary area that brings together researchers and applications from several other related areas such as Music, Computer Science, Information Visualization, Design, and their specialties. Thus, this large area can be categorized by its types of applications, such as symbolic visualizations focused on the study of structural and elementary information in

music, which can be analytical as in Cabral [2], animated and interactive as in Cantareira [3] and Silva [4] communicating musical data in real-time. They can also be used as a tool for music education Hein[5] and creation of compositions [6]. The category of sound visualizations, which focuses on the audiovisual processing Jacomé [7] and Scordato [8] which develops Ianixx, for the creation of artistic performances.

Performing exploratory research on the Music Visualization area, we synthesized its definition with the diagram shown in figure 1.

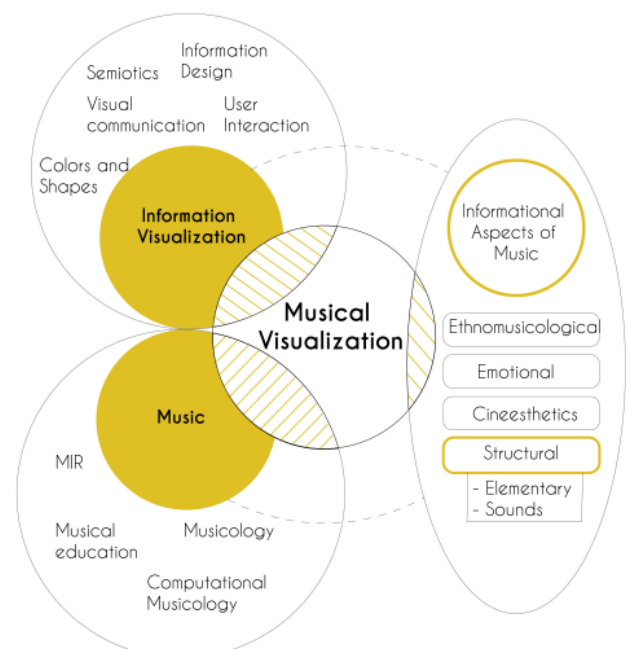


Figure 1: Visual Music

Given the breadth of application of musical visualizations in various areas, we consider it as an intersection and, through new technology, it allows to improve communication, study, and understanding of music as information and as a sign [9]. The MIR (Music Information Retrieval) area, for example, has well-established works in the context of developing a set of tools for computationally extracting musical information, such as Music21 [10]. However, the development of musical visualizations still has limitations, mainly in relation to multi-representational challenges, Futrelle [11]. Considering that this is an area that is still maturing, we seek to carry out studies in relation to different ways of representing music. Therefore, we emphasize that to create a musical visualization, we must consider its informational aspects, that is, which musical information will be visually communicated.

*Supported by FACEPE.

[†]<https://music.cin.ufpe.br/>

3. Related Work

Over the last few decades, several works have been deepening research in areas such as Music Information retrieval (MIR) and Semantic Audio Analysis (SAA), whose objectives are, respectively, to retrieve information from musical data and analyze information in audio [12].

An example of relevant work in this area of study was the development of Pure Data, the free visual programming software for computer-supported interactive music works developed by Miller Puckette [13], which expanded the possibility of experimenting with data processing computational music.

Based on Miller Puckette's work, LibPd was created, a library that makes Pure Data an embeddable tool in any [14] application. Since then, many works started to use Libpd to create and manipulate musical and audio information, making Libpd one of the most used libraries in the area by authors such as Wilcox[15] Brinkmann [16] Konrad [17] Hathway [18] and Steps [19].

More recently, other libraries and toolkits have been explored for such work, such as MatLab, well-known mathematical analysis software that has developed a toolkit for MIR [20].

Another great revolution was the emergence of the Musical Instrument Digital Interface (MIDI), which made musical communication possible with the robust [21] protocol. This protocol has evolved over the years allowing for better communication between instruments and computers. Recent works try to facilitate the use of this tool through a better human-computer interface for creation and interaction with MIDI Machado [22] and Amorim [23]. Another example is applications that help experimentation and create new rhythms from rich interfaces through the MIDI protocol, as in Wilson [24].

On the other hand, interdisciplinary studies deepen discussions in the Musical Visualization area, which carry out research to make musical information more visual and understandable to the eyes. As in Vieira's work [25], which uses dynamic visualizations to understand tones with parameters of graphic animations, with variations in volumes and colors according to the musical energy, also seeking to realize emotional aspects such as warm colors are attributed to joyful moments and cool colors for tragic music moments. The *Musanim* [26] is also a practical example of dynamic visualization, as is the *Da Capo* [4] which uses MIDI symbolic data inputs, to allow interaction with controllers through controllers musical instruments generating musical visualizations. Or even in tools like MusicVis [27], which through the *input* of MP3 files or microphone capture performs the processing of musical data generating a real-time visualization marking the visual elements in the graphic space of the browser. system user, analyzing the entire musical sequence according to timbres and sound frequencies.

Considering all this state of the art, this article tries out the use of some technologies found in related

works, such as Libpd, and uses symbolic data input from MIDI files to explore musical information generation and visualization in interactive applications.

4. Developing Music Visualizations

In this article we will describe the development of two software, *Espectromusic* and *Mandrit*, which we developed through the processing of musical data, with the extraction of MIDI file characteristics and transformation of its information into structured formats to obtain dynamic and static visualizations that visually communicate musical and sound information through graphical results.

4.1. Espectromusic

The *Espectromusic* is a multiplayer game in which the player aims to overcome a digital labyrinth created and manipulated in real-time by music captured by a microphone. Creating a tournament dynamic, as shown in the figure 2 with collections performed during user validation processes. Engaging interaction and competitiveness between the listener and the opposing musician. And it can also be used as a complementary projection to the musical performance according to the variation of volumes and frequencies of what is played and which forms the spectrum of obstacles.



Figure 2: Validations Espectromusic

Designed as a platform game, it has two main scenes, the first with audio input, in other words player 1 interacts with a predetermined song, as a tutorial stage interacting with the *ball* through the keyboard controls. aim to cross the obstacles until the next stage. In the second phase, you can interact directly with player 2, the musician, considered the generator of obstacles when playing the music, raises the spectrum and with his instrument can build obstacles preventing player 1 from moving forward with his *ball*.

4.1.1. Extraction of sound characteristics

When we developed the entire graphical interface of the game by Unity, we received the musical data captured in

real-time by the microphone and transformed it into dynamic visualizations with a graphic area determined by the elevation of bars according to the sound spectrum, having as input what the musician instrumentalist plays.

To process the sound data captured via the microphone, we used the *LibPd* library as shown in the flowchart of figure 3, which allows input and control over the sound, transforming Pure Data (PD) into an embeddable library in the Unity IDE. PD allows you to process raw audio from audio drivers and MIDI drivers, which makes it possible to use musical and sound information in mobile phone applications, game systems, and web pages.

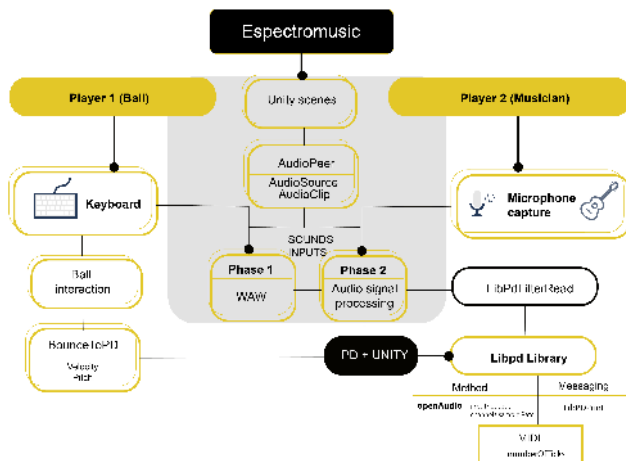


Figure 3: MIDI input microphone processing

Therefore, it was necessary to develop a unity script that uses a fundamental frequency and adds to the variation of the ball's speed in the scenario, taking the pitch of a certain frequency and sending it to the scene through LibPd, in a cyclic exchange that passes it on to the PD to synthesize in real-time. For this real-time integration of Unity and PD to occur, the programs must be open synchronously for the game to run.

We use the *audiopeer* methods that extract values from the spectrum by calling the *audiosource* script, which is the component that reproduces the music, in the first phase in WAV format, and in the second phase, we obtain musical information captured via a microphone, and apply to the objects of the game scene. In addition, *audio-clip* complements the processing, having the function of storing a sample of audio data, which has variations in the spectrum generated about intensity, referring to the amount of energy, height according to the format of the sound oscillations, and timbre determined by frequency, with this information being shown at each frame and generating obstacles.

It is important to note that the LibPd library works internally through MIDI messages, such as note-on, integer values of the channel, pitch, and velocity. And it sends a control change event to the PD. So, we have a script called *LibPdFilterRead* that opens the microphone door when we enter the second phase of the game and initializes PD by taking the Unity sample rate and calculating

the number of ticks.

We emphasize that during the development of *Espectromusic* we also explored audio processing, to apply reactive sound effects to the *ball* object which the player manipulates according to speed variation, creating sounds through visual programming with PD, as shown in figure 4.

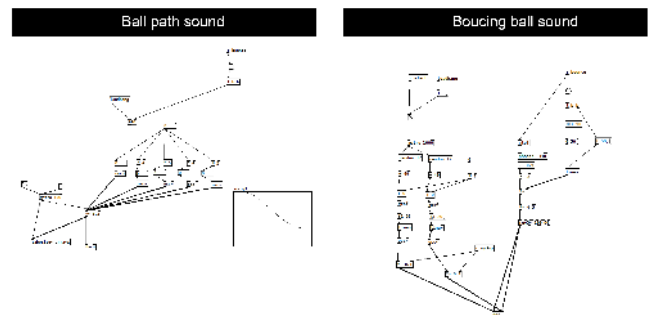


Figure 4: Visual Sound Programming with PD

In the PD graphic environment, we create sound objects, called "patches", we apply two characteristics to *ball*, in (I) the sound of its path, triggered by a certain MIDI frequency, with speed and height controls, which vary according to input values determined by the oscillators. As in II, the sound of the *ball* bouncing, simulating its touch on the game's floor plan, but with random variations of frequency values.

These files synthesize a continuous and pleasant sound according to the received MIDI frequencies, and with the use of the converter *dac* they transform the digital to analog, generating a sound output. It is important to emphasize that this PD graphic environment format for sound creation made it easier for the project developers to carry out the manipulation of digital audio data, experimenting, listening, and visualizing sound waves, spectra, frequencies, and amplitudes.

4.1.2. Challenges in data interference

With the development of *Espectromusic*, we faced some computational challenges about audio interference in raw data capture, due to microphone transmission and real-time data acquisition, which depending on the environment also transmits noise or information that we had no computational control. In addition to the deviation of data by the PD inputs, due to the use of multiple inputs, both from a complete music file and the sound applied in the interaction of player 2 with the *ball* having variable pitches according to the speed of the movement.

In addition to these aspects of musical data, we highlight the need for tools for musical game development that perform dynamic integration and manipulation between the Unity IDE and the PD. Because despite the LibPd library, which has been a great alternative in our case, it still had several computational limitations, for example, the game only works with Unity3D's 32bit editor.

Another difficulty was to develop sounds that were pleasing to the ears by manipulating the oscillators and perceiving according to the speed of the ball. Needing a thorough job of defining the ideal frequency and the limits of variations, as well as synchronizing the sending of data in real-time. So we consider that there are also gaps about the domain of tools and technical specifications in the choice of sounds, in the attribution of geometric shapes, as we need to carry out in-depth studies in the manipulation of the 3D area and construction of the scenario. That's why it made it easier to have specialist members in Music, Design and Computer Science accompanying and developing the project together, and thus we obtained visual and interactive results with *Espectromusic*.

4.2. Mandrit

Knowing the various applications and potential of software resources - for processing, processing, and manipulating data from digital MIDI files -, we developed the *Mandrit* system: musical visualization generator for rhythm analysis, as shown in figure 5, it draws three types of graphic models to visually communicate rhythm and demonstrate its micro information through the rhythmic signature of a musical work.

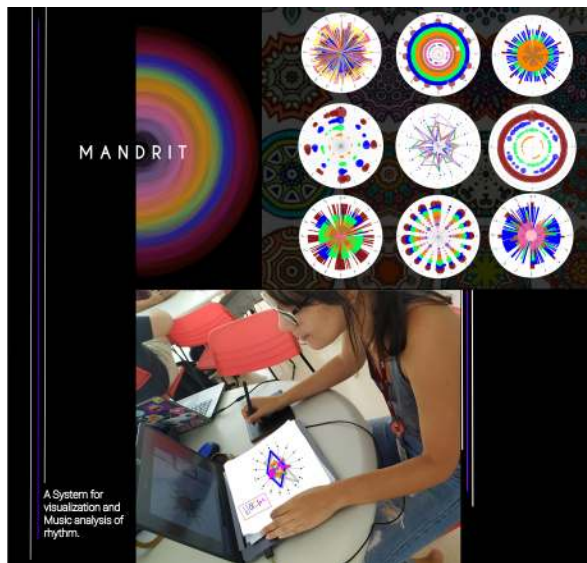


Figure 5: Musical Rhythm Analysis

With its visualizations, *Mandrit* proposes a musiological study of musical analysis processes and Auditory Scene Analysis (ASA)[28] to measure musical perceptions with the aid of graphical representations, using qualitative and quantitative assessments.

With the collections of feedbacks, we realized as well as [29] that the results show that direct associations of musical perception by the association of colors as a representative entity of the instrument, and also the participants make correlations based on graphic visual structures that are largely determined by context, and their abilities to understand and perceive music associating musical visualizations.

The elementary information of the song chosen to develop visualizations was the rhythm. Realizing the scarcity of analytical and computational research for the study and analysis of rhythm, its micro information that visually communicates variations, musical signatures, polyrhythms, differences, and similarities of musical works.

4.2.1. Extraction of rhythmic characteristics from MIDI

In this case study, the process of exploring and choosing rhythmic information to generate a musical visualization is carried out through tutorials and collection of feedbacks with specialists in music and the area of Graphic Expression and Design. Therefore, we emphasize that to build a musical visualization, we first need to define the properties that we want to communicate visually according to the needs of potential users. Therefore, with *mandrit* aiming to generate visualizations that emphasize the signatures of the music time signature. Seeking to facilitate the analysis of intervals, pulses, patterns, and regular or irregular movements of bars. Therefore, we performed the extraction of these characteristics through the mandrit algorithm, described by the flowchart of figure 6. The rhythm signatures, which we generate, are based on a volume of data extracted from MIDI files, extracting the events of the measure, its metrics, and counting the number of notes of each instrument.

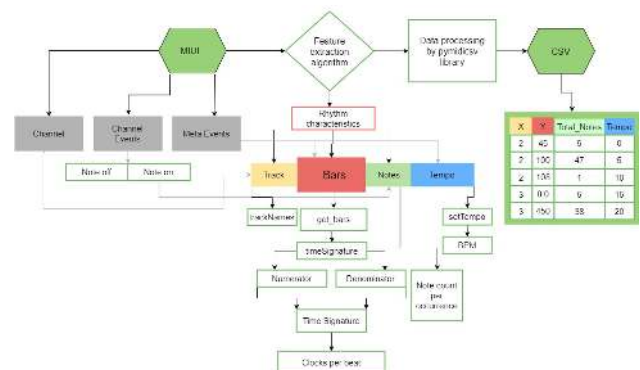


Figure 6: Fluxograma Mandrit

With it, in addition to performing the specifications of the characteristics of the rhythmic element, obtaining its metadata, we also created a database with a collection of over 60 songs, different genres, and with simple metrics and enabling the analysis of bars. We structure and store musical sequences according to track variables, time signature, and amount notes. To develop the extracted code, we determine tasks to develop with the following steps:

- (1) Read MIDI file in type 01 format.
- (2) Declare measure data for each song with the `get_measure` function.
- (3) Extract meta-event from bar signature.
- (4) Calculation of bar division
- (5) Count the amount of "Note on" notes.

- (6) Generate the matrix with three elements: X (tracks), Y (Measure subdivision) and Total_of_Notes (Amount of notes).

To carry out these specifications, we access the raw data of a musical sequence, and collect the information from the *tracks* by accessing the channels of the MIDI file, counting the occurrences of the number of notes only by the *noteon*, and considering respectively each temporal occurrence as a function of the measurement metrics. The transformation of these raw data and metadata was done through the library Py_Midicsv [30] which has instructions determined by the type 1 format protocol of a MIDI file, in which all data are separated by track. With this implementation, we store data in a spreadsheet format with structured and organized information, facilitating the generation of *Mandrit* views. We apply mathematical and computational functions aided by software resources, developing the algorithm described in figure 7.

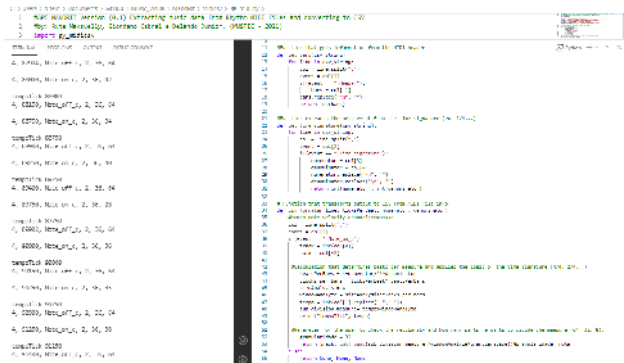


Figure 7: MIDI Rhythm Extraction

We emphasize that we extract only some information from the MIDI file, making specifications for the elementary rhythm data to be returned, the parameters of "tracks" are also called tracks, the signature of the measure and "Notes". In other words, these were our fundamental variables to store in numerical data of the musical sequence and represent a synthesis of the rhythm of each musical work to be analyzed.

Based on the structured data generated by extracting rhythm characteristics, we also developed an algorithm that generates the *Mandrit* graphics, as shown in figure 8. Where we assign rhythmic information applying to graphic shapes - bubbles, lines, dots, colors -, in addition to using polar mathematical functions, that is, it has an angular orientation according to a given rotation.

The generation of static graphs was the result of tests seeking to assist in the priority display of the amount of notes as a function of temporal occurrence, so *mandrit* generates three types of graphs: Polar Bubble Chart, Polar Radar Chart, and Radial Collumn. All of them have the same principle in the representation of rhythm in a metaphor to clockworks with their polar and cyclic visualizations, however, we experience in their visual approaches the variations of graphic forms attributed to musical information.

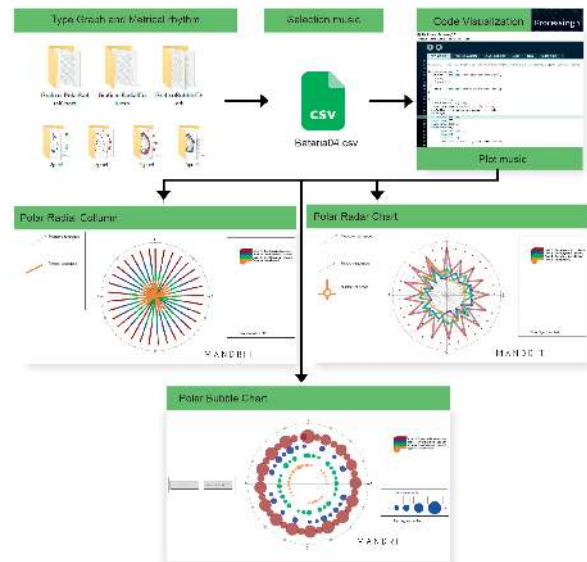


Figure 8: Plotting rhythmic visualizations

We considered a set of fidelity tests performed accompanied by visualizations, which were very useful in the process of understanding and perceiving the organization of musical data in the representation. The expert contributors, in addition to bringing provocative *feedbacks* and qualitative impressions, also contributed to the musical analysis by noticing elements that were not visible before, such as the similarities between aspects of musical performance about the instruments of a musical work.

4.2.2. Challenges generations of Mandrit Views

In every experimental process of constructing musical visualizations, we face computational challenges about the treatment and extraction of rhythm characteristics. A first limitation is that *Mandrit* only extracts rhythmic data from MIDI files in type 1 format, due to the data structure we need to have the tracks organized and separated to facilitate musical analysis. However, because of the productions available online, we often find files that complicated the processing and resulted in graphic pollution, as in the figure 9.

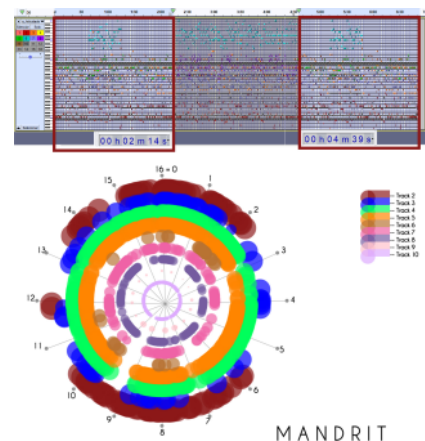


Figure 9: MIDI production challenges

When we plotted the music, we considered the amount of accumulated information incomprehensible, and when we checked the musical production, we realized that the producer gathered excerpts from the song "A Felicidade, Toquinho" in its initial and final parts in conjunction with Bossa style randomness, which resulted in a file of 06 minutes and 53 seconds, with a complex and accumulative amount of musical data. For this reason, the plotted view does not show the rhythm signature correctly demarcated, and it also has no reference to its instruments in the legend, which limits the reading and communication of the rhythm.

Still, on the aspects of independent productions, we highlight that we often find files without track descriptions, or their musical sequences completely disorganized. This made it difficult to plot the views, even to apply parameterization and generalization of graphic models. On the other hand, all this also demonstrates the musical artistic singularities and complexities. Like for example in the song "Take five, Dave Brubeck". It has a complex 5/4 time signature and so we experimented with plotting in different arrangements, as shown in the figure 10 and soon noticed inconsistencies in reading files from different MIDI bases.

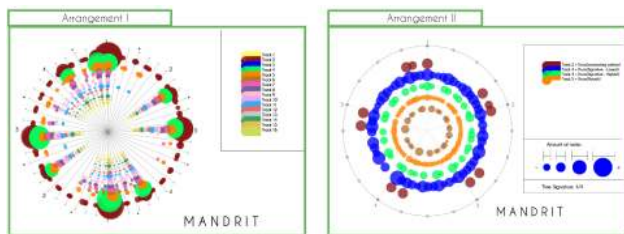


Figure 10: Arrangements

As we observed in "Arrangement I", it generated a more complex visualization with a greater number of tracks (tracks) being incomprehensible its fundamentals since it is a piece of quinary music, that is, with a complex five-pulse bar. In the MIDI file that we printed the "Arrangement II", with the information much more organized and structured, it is possible to obtain this signature, through the improvements we made implementing a generalization in the length of the song according to a granularity applied to the values in the extracted data, which can range between 64, 32 and 8 in granularity.

That is, these values can be changed by the user who generates the visualization, depending on what they need to conceive or understand in the musical analysis of the work.

Through the oscillations, and variations in the counting of the number of notes demonstrated during the course, we understand that these differences in the aspects of expressiveness are notorious with different productions and files of the same music. This context is explained by the music production process, which is relative according to each musician, demonstrating in the different graphic results their distinct expressiveness, despite retaining a sig-

nature in the most striking poles of the rhythm.

There were also some limitations regarding the generation of static views, due to graphic pollution resulting from the accumulation of temporal information plotted in a two-dimensional graphic area. That's why we bring modifications in the reference of the tempo with the grid to the movement and orientation of the clock and a metaphor to the rhythmic time represented by an animated cursor as its pointer to help in the accompaniment. In general, there are different challenges and specific gaps that arise during the process of extracting information from raw data, and there are also several possibilities to visually represent and communicate information with of this, we learn the advantages and disadvantages of using certain graphic forms and to make choices according to each need, always adapting to the objectives centered on the users.

5. Guidelines for developing Music Visualizations

As a result of these case studies and in the development of *Spectromusic* and *Mandrit*, several lessons were learned about the process of building music visualization systems, mainly regarding the selection, extraction and generation of representation. With the creation of prototypes we were able to obtain musical parameters, visual structures and identify challenges and limitations to visually communicate musical information.

Thus, in addition to the technical records described in this article, we also provide guidance on the challenges of exploring this interdisciplinary area, which is Computational Visualization Music. Designing 5 guidelines that can help in the process of creating a system to generate musical visualizations, they are:

- (1) Conducting similar research, user-centered technical and participatory observation to explore musical information to be communicated visually.
- (2) Categorization and selection of visualization types to develop.
- (3) Exploration of the creative process aided by computational resources, to generate automatic visualizations through the extraction, treatment, filtering of musical data and application in *visual structures*.
- (4) Use of a Design process based on cyclical processes of ideation, prototyping, and evaluation with *feedbacks* collection and fidelity tests.
- (5) Communication and sharing of graphic results to strengthen the community.

We consider these to be crucial steps to achieve the objective of visually communicating musical information, through we built an experimental protocol model based on *Design Thinking* composed of targeting cards and methods applicable to graphic experiments, enabling a systematically mapped view of automatic generation of musical visualizations. In addition to broadening the discussion, and unraveling gaps in the quest to facilitate and com-

plement communication between professionals and experimenters in the field of music. Transforming computer interfaces and tools as a means to make musical information accessible and understandable.

6. Final Considerations

With these experience reports, we highlight the importance of interdisciplinary research as a means to expand the application of technology in the music field, narrowing the limitations and connecting professionals from fields that can complement each other, generating useful and interesting results. For this, it is necessary to emphasize that the follow-up with specialists in the area of music and Information Visualization, in continuous cyclical evaluation processes with fidelity tests, were essential to achieve graphic results and obtain improvements in each session and experimentation round.

Despite this, communication between developers and musicians is still a gap in the extraction and transformation process in prioritizing data to be visually reported. Because there are technical dependencies required due to the need for music theory studies to flow this exchange.

Acknowledgement

To the Collaborators of the Mustic research group: Ricardo Sholtz, Delando Júnior, Flaviano Dias, Felipe Calegário e FACEPE.

References

- [1] José Fornari. Da música à musicologia, January 2019.
- [2] Giordano Cabral and Robert Willey. Analyzing Harmonic Progressions with HarmIn: the Music of Antônio Carlos Jobim. page 12, 2007.
- [3] Gabriel Dias Cantareira, Luis Gustavo Nonato, and Fernando V. Paulovich. MoshViz: A Detail+Overview Approach to Visualize Music Elements. *IEEE Trans. Multimedia*, 18(11):2238–2246, November 2016.
- [4] Mateus Bastos da Silva. Da Capo: Uma exploração das representações da música, 2019.
- [5] Ethan Hein. *Designing the Drum Loop: A constructivist iOS rhythm tutorial system for beginners*. PhD thesis, Steinhardt School of Culture, Education, and Human Development, New York University, 2013.
- [6] Delfina Malandrino, Donato Pirozzi, and Rocco Zaccagnino. Visualization and music harmony: Design, implementation, and evaluation. In *2018 22nd International Conference Information Visualisation (IV)*, pages 498–503, Fisciano, Italy, July 2018. IEEE.
- [7] Jarbas Jacomé. Sistemas Interativos de Tempo Real para Processamento Audiovisual Integrado, 2007.
- [8] Julian Scordato. Composing with Iannix. In *Proceedings of the Fifth Conference on Computation, Communication, Aesthetics and X.*, volume V, page 389, Lisboa, Portugal, 2017.
- [9] Eero Tarasti. Los signos en la historia de la música, historia de la semiótica musical. *Semiótica Musical*, pages 15–71., 2008.
- [10] Michael Scott Cuthbert and Christopher Ariza. music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data. page 7, 2010.
- [11] Joe Futrelle and J. Stephen Downie. Interdisciplinary Research Issues in Music Information Retrieval. *Journal of New Music Research*, 32(2):121–131, June 2003.
- [12] Bozena Kostek. *Perception-based data processing in acoustics: applications to music information retrieval and psychophysiology of hearing*, volume 3. Springer Science & Business Media, 2005.
- [13] Miller Puckette et al. Pure data: another integrated computer music environment. *Proceedings of the second inter-college computer music concerts*, pages 37–41, 1996.
- [14] 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*, volume 291. Citeseer, 2011.
- [15] Dan Wilcox. Pdparty: An ios computer music platform using libpd. In *Proceedings of the Pure Data Convention*, 2016.
- [16] Peter Brinkmann, Dan Wilcox, Tal Kirshboim, Richard Eakin, and Ryan Alexander. Libpd: Past, present, and future of embedding pure data. In *Proceedings of the Pure Data Convention*, 2016.
- [17] Markus Konrad and Klaus Jung. *Analysis of audio synthesis possibilities on mobile devices using the Apple iPhone and iPad*. PhD thesis, Ph. D. Thesis, HTW Berlin, 2011.
- [18] Joe Hathway. Creating a digital musical instrument for children’s education and expression. 2021.
- [19] Leonardo Porto Passos, Leonardo Arruda, and José Fornari. Pure data for pure audio games.
- [20] Olivier Lartillot, Petri Toivianen, and Tuomas Eerola. A matlab toolbox for music information retrieval. In *Data analysis, machine learning and applications*, pages 261–268. Springer, 2008.
- [21] David Miles Huber. *The MIDI manual*. Sams, 1991.
- [22] André Machado. *Tradutor de arquivos MIDI para texto utilizando linguagem funcional CLEAN*. PhD thesis, Universidade Federal de Uberlândia, Uberlândia, 2001.
- [23] ANDRÉ AMORIM and JOHANNES KJELLBERG. A step out of the grid-interaction with touch-based alternative rhythm programming layouts in drum machine user interfaces. 2020.
- [24] Scott Wilson. Patterning: The iPad drum machine that wants us to think in circles, 2015.
- [25] Marco Filipe Ganança Vieira. *Interactive music visualization: implementation, realization and evaluation*. PhD thesis, 2014.
- [26] Stephen Malinowski. Music animation machine renderers, 2012.
- [27] Van Huynh and David Comberg. MusicVis a web application for visualizing sound., 2017.
- [28] Albert S. Bregman and Stephen McAdams. *Auditory Scene Analysis: The Perceptual Organization of Sound*. *The Journal of the Acoustical Society of America*, 95(2), February 1994.
- [29] Gregor Strle e Matev vz Pesek e M. Marolt. Towards user-aware music information retrieval: Emotional and color perception of music. In *Emotions and Personality in Personalized Services*, 2017.
- [30] John Walker. MIDICSV, 2004.

A preliminary study of Augmented Musical Instruments for Study (AMIS) using research through design.

Eduardo Santos Silva^{1,2*}, Giordano Ribeiro Eulalio Cabral^{1,2}, Rodrigo Mendes de Carvalho Pereira²

¹ Centro de Informática – Universidade Federal de Pernambuco
Av. Prof. Moraes Rego, 1235 - Cidade Universitária, Recife - PE, 50670-901

² Daccord Music Software
R. Vig. Tenório, 194 - Sala 401 - Recife, PE, 50030-010

ess@cin.ufpe.br, grec@cin.ufpe.br, rodrigo@daccord.com.br

Abstract

The emergence of Digital Musical Instruments (DMIs) in the computer music field has been providing new means of interaction with music performances. Particularly, with the advancements in the Internet of Things (IoT) area, there has been an increase in augmented musical instruments containing LEDs within their own bodies to support music learning. These instruments can help musicians by having the capabilities to display musical notes, chords and scales. This research uses research through design to present a preliminary study of some of the challenges related to the design of these instruments and attempts to provide some insights associated with their usability and development, particularly related to the development of an augmented acoustic guitar, the VioLED, developed alongside the company Daccord Music. The system provides three modes of operation: song mode, solo/improvising mode and animation mode. The preliminary results present challenges and obstacles pertinent to usability and development of these instruments identified in different iterations of design. These challenges are associated with areas such hardware-software co-design, usability, latency/jitter, energy consumption, etc.

1. Introduction

In recent years, the computer music area has witnessed the emergence of several technological advancements that allow new means of audio and musical interactions. Applications that involve from new ways of song playback [1] to new Digital Musical Instruments [2] and smart instruments [3]. Particularly, in the field of gestural interfaces, the technological advancements enable the development of new musical instruments augmented with sensors, actuators and microcontrollers to provide new features to the user, also known as hyperinstruments, to extend their sound capabilities [2].

1.1 “The perfect storm”

In the last few years, the IoT research area has gained attention with the increasing popularity of mobile consumer electronics such as smartphones, tables, wearable devices, etc. The increasing number of devices

connected to the internet [4] and the increasing accessibility of broadband mobile internet connection, microcontrollers, e.g., Arduino, Raspberry Pi, wireless communication technologies, e.g., Bluetooth, Xbee, NFC, as well as sensors and actuators, produce a scenario that can stimulate the development of new electronic projects. Simultaneously, the growth of music streaming services, e.g., Spotify, Apple Music, Deezer, Tidal, etc. provide expanding content to new music applications that vary from IoT Playback devices, e.g., Aumeo, audio customization, setups for collaborative creation, devices for artist-audience interaction, smart instruments, etc. [3].

1.2 Augmented Musical Instruments for Study (AMIS)

In the context of musical learning, there has been an appearance of instruments a particular set of augmented instruments, referred in this paper as Augmented Musical Instruments for Study (AMIS), which use light-emitting diodes (LEDs), projection and other means to display musical content directly on the instrument’s body as means to help beginner musicians.

Several researchers and commercial products have focused on augmenting musical instruments to support music learning [5, 6, 7, 8, 9, 10, 11, 12]. These systems can display musical notes, chords, scales and the vast musical content and body of technique, developed across generations, by using different approaches. A few of the systems use highly intrusive approach by implanting LEDs and Print Circuit Boards within the instruments manufacturing process [5, 8, 9, 10]. Other systems still utilize LEDs, however, exploit a non-intrusive approach by creating a “sleeve” that can be inserted over the instrument [6]. Other researchers use projection systems to display the musical content over the musical instrument. These systems [10, 11] use software to convey notes and chords to the user. Furthermore, there are approaches that use the Head-Mounted Displays to support music learning, e.g., Music-Everywhere [12].

1.3 Applications and services

Different applications and services can provide musical content and tools to the learning process. Musicians often use tablature and chord charts services

* Supported by FACEPE

such as Ultimate Guitar, Cifra Club, Guitar Pro, Songsterr, etc. in the learning process. These provide musical content with different music notations (sheet music, tablatures, chord charts). Video sharing websites, e.g., Youtube, have become more popular among musicians as a source of tutorials created by the users. Furthermore, software such as Yousician, Synthesia, Chordify, Yalp, etc. provide interactive content to support music learning also emerge with different approaches incorporating social network and gamification elements.

2. Challenges in the design process of the AMIS

These AMIS exist in a very particular design context that present obstacles in different areas: HCI, Real-Time Applications, Hardware-Software co-design and music learning.

2.1 Challenges in HCI

Some researches debate that the role of interaction design is to provide usability to the design process by making systems safe, reliable, easy and pleasant to use at the same time that they are functional [13]. The challenge is to provide usability according to different aspects that can affect the system, such as organizational aspects, environment, safety and security, user engagement, comfort, task related aspects, user interface, among others. Further details about the criteria, the different stakeholders involved in system's usability is present in the literature [13].

The research of Dix et. al. [14] and Nielsen [15] provide insight on different criteria to improve usability. However, the challenge remains in identify the trade-offs between each system's priority features and their effect on the overall usability. As an example, the vast number of features in Adobe's Photoshop may be very helpful to users, but the amount of content can increase the intricacy of its user interface and might cause an overwhelming sensation at first.

Moreover, the evaluation of these criteria is somewhat complex, involving qualitative research with specialists and end-users. Some researchers discuss that for mobile applications, interactions design has some limitations related to different devices' components, screen size and navigation structures [16]. Particularly in the case of the AMIS, the systems and services for music learning have real time synchronized interactions, a variety of different complexity content, and interaction elements that can influence the usability and how these elements can affect the learning experience.

2.2 Challenges in hardware/software co-design (HW/SW co-design)

Predominant in development of integrated circuit systems, especially in embedded systems, research in the literature involves analyze how to address system requirements through hardware and software synergy [17].

Several challenges in HW/SW co-design refer to

management and tools for different teams working in parallel, specific tests, etc. For real time applications, requirements in performance can be greater than control systems. However, control systems deserve greater attention in reliability, safety/security, etc. Some authors highlight three aspects during the design process: modeling, validation and implementation [17]. In IoT projects, these challenges emerge when designing new systems, since most specific and proprietary applications need custom hardware and software. Moreover, interactive systems connected to mobile devices often need to assess which features are better implemented in hardware and the ones are more suited for software. This is a particular challenging design problem when dealing with time-to-market [18].

Furthermore, HW/SW co-design can be a challenge in the prototyping phase. For systems that lack flexibility regarding changes and improvements in hardware, it can hinder both exploration within the design space as well as the freedom to meet new software requirements [18], therefore, further delaying the design process.

2.3 Challenges in real time applications

Real time systems depend not only from the information and system behavior, but also from how long it takes to the information to arrive or the functionality to occur. These systems highly demand from precision and reliability on the information and system's response, since any errors or delay can have fatal safety and security issues or completely interfere in the user's experiences.

DMIs and AMIS also have the real time requirements since "time is central feature in music" [19]. Therefore, errors related to package loss from the communications module or latency can hinder the experience by showing incorrect notes at incorrect times. Latency and jitter are common challenges when designing DMIs and therefore AMIS [20].

The prototyping phase of AMIS can also be challenging because deals with tasks that increase in complexity. Similarly to the game development process, it becomes challenging to envision all the functionalities and elements involved in the experience by paper prototypes or models [22].

2.4 Challenges in music learning

As discussed in previous sections, both hardware and services associated present challenges in different areas during the design process. However, these services aim to support music learning therefore on must consider aspects related to informal/formal learning, types of music notation, learning new abilities and their learning curve.

Learning a music instrument involves acquiring a whole body of technique developed across generations [23]. Both cognitive abilities to recognize and understand music notations and structures [24], but also motor skills to apply these music concepts into the instrument [19, 21, 24].

Some authors discuss different types of musical learning processes. While formal learning focus on

understanding music notations, and the body of technique of that instrument [23], training virtuosos [25], informal learners use video lessons, friends, family, simpler music notation (tabs and chord charts), popular repertoire and focus on “play music” instead of “how to play and compose music” [26].

The graphical notation also can influence the understanding of the musical concepts. While sheet music provides more complex and deeper understanding of all aspects related to notes, music structures, it may become cumbersome for beginner musicians to learn at the same time they need to acquire motor skills to apply these techniques. In contrast, tablatures and chord charts provide a simpler notation to understand music to the detriment of some musical aspects, such as time of execution of each note and chord.

Some research discuss that the learning curve of musical instruments relates to the time taken to achieve certain level of skill to master the instrument or its body of technique [27]. Some authors argue that it can take up to ten years to achieve this point [19]. Other research discuss that the instrument needs to provide easy access to understand and achieve results at the same time they provide enough challenge to master [28].

3. Method

This section discusses the method and design tools used in this research.

Zimmerman et. al. discuss that the research through design method emerged as tools for designers to handle complex systems and “wicked problems”, i.e., problems that are too difficult to model because of conflict between its stakeholders and aspects [29]. Other authors use the term “design-oriented research” to define the use of design methods to produce academic research and knowledge [30] by developing, testing and evaluating prototypes. Prototyping is an essential step when de designing new systems, as each iteration containing tests and evaluation of the prototype can reveal flaws, features that need to be removed and essential requirements, reducing workload in non-essential features and guiding the design process.

The method used in this paper follow an iterative process containing three steps: learn, build and measure (Figure 1).

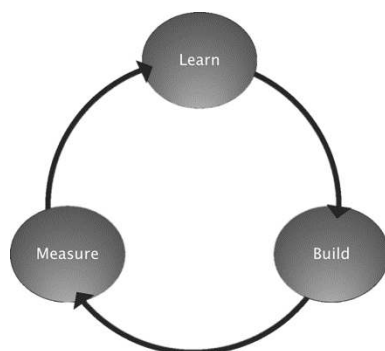


Figure 1: Design method

The learning phase involve understanding the context, collecting interviews and reports about the about

the subject, studying market trends and analyze related work in the literature. The build phase involved using design tools to produce a prototype containing basic essential features for evaluation in the measure phase. This step is essential to identify the system’s requirements. The methods applied were Brainwriting 6-3-5 and a modification of Crawford Slip. The modification on the Crawford Slip method was integrating sketches in idea elaboration so the participants could elaborate how the user interaction would be. Furthermore, prioritizing ideas and specifying requirements based on innovation, target audience, market strategy, music learning and technological aspects. In the measure stage, the internal team and potential users test the prototype to evaluate which features need to be discarded and which features advance to the next iteration.

3.1 Learn phase

After reviewing literature research, the next step was conducting interviews to identify key aspects as well as gather the design team to perform the brainwriting and Crawford Slip methods. Because of the ongoing COVID-19 pandemic, the method for conducting interviews was through Skype and Zoom calls with 27 subjects between March and May 2020.

The subjects vary between beginners, professional and aspiring musicians, as well as people who previously tried to learn guitar. The interviews tried to assess their experiences with music learning, discussions about learning methods and possible features that were lacking when they learn (or attempted to learn) the instrument.

Analyzing the interviews, the subjects discussed key aspects: learning chord positions, feedback, improve technique, freedom to learn in mobile devices and finding song within their level that they wanted to learn. Some interviewers described that their learning experience with professional tutors or schools were mainly formal learning and that discourage them to pursue further learning or that they “endure” the process. One subject reported that they wanted to “first learn how to play, theory comes later”.

When questioned about how do they thought music learning was going to be in 15-20 years, most described learning through mobile devices, but still respecting the tutor’s role. Three subjects even reported the possibility of the emergence of “devices connected to the instrument to see chord charts”, “an augmented guitar that shows chord charts” and “a guitar with some kind of holographic fretboard” to “see the chords in the fretboard”.

3.2 Build phase

Before starting building the prototype, a team containing six individuals between 20 and 40 years old from different departments (three from management department, two from development and one from marketing) participate in the brainwriting 6-3-5 and variation of Crawford Slip methods.

The results identified five key features: legibility, integration with vast song databases, social network, gamification elements and simple design. In this context,

legibility relates not only to the display format on screen (size, font, etc.), but also to the understanding of the content itself. That way, the music notation chosen at first was the chord charts notation. Integration to vast song databases were related to associating the musical learning with music streaming services, which led to synchronizing music learning with song playback. Additionally, Elements in social network and gamification can improve user's motivation, which could help users that previously tried to learn the instrument and lost interest or aspiring musicians who look for an informal learning style.

4. The VioLED prototype

The VioLED (Figure 2) is an acoustic guitar containing a PCB board with LEDs behind the instrument's fretboard, developed in a partnership between the companies Daccord Music and Batebit Artesania Digital. The company Batebit developed the hardware and performed all its changes across versions, as well as the prototype's version 0.1, working alongside a luthier to insert the electronics into the instrument.

The prototype aims to benefit from IoT technologies to support music learning. The development of the prototype also resulted in two patents related to the instrument itself (BR1020170083071) and its communication protocol (BR1020170082962).



Figure 2: VioLED.

All versions of the prototype use a screwd-in PCB board containing 78 WS2812B RGB individually addressed LEDs. Each LED represents one note in the fretboard up to the 12th fret, as well as the open strings with light diffusers in each hole.

The prototype contains four different parts: hardware, control app, embedded software and communications protocol. The controller is an Arduino UNO board that sends data to control the LEDs features: on/off, brightness, color, etc. To perform tests in hardware, both the Arduino and the power supply reside inside an external mdf wooden case constructed using a laser cutter. A modified XLR cable powers the PCB board and sends the data to control the LEDs. On the other hand, the control app varies in different versions of the prototype, from desktop apps, to Android and iOS apps. The control software sends the notes and chords to an Arduino UNO board containing the embedded software.

4.1 Low fidelity prototype

At first, as discussed in previous sections, the company's team participate in design methods to generate ideas for the ideal control app. Each participant tried to

illustrate features and basic user interactions (Figure 3).

4.1.1 Discussion

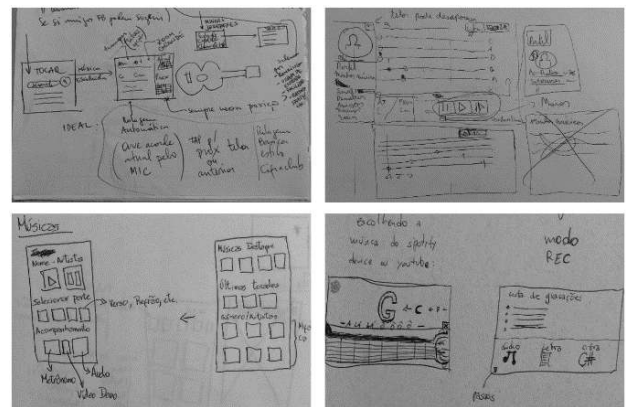


Figure 3: Low fidelity prototypes.

During this phase, the team realized the challenge in imagining the complete user interaction. The different possibilities in device orientation, number of elements displayed, highlighted elements and in deciding elements on screen and on hardware.

The main concept in this phase was the need to anticipate the chords to the user. Analysis of music learning methods and similar software like Yousician, Chordify, Guitar Hero, Guitar Pro, etc. revealed that all music notation provide a way to anticipate which notes and/or chords the user has to play. This helps the user in adjusting their posture, hand and fingers position as well as developing a mental plan of the music structures involved.

4.2 Version 0.1

In this version, the control app developed in Processing had a graphic checkbox interface to control each LED's I/O state and powered by a power supply. The system sent messages over serial protocol to the Arduino board.

4.2.1 Discussion

The possibilities in controlling each LED, its color and brightness presented way to display notes chords and music scales. However, the interaction was too cumbersome, as the user needed to select each individual note. Therefore, it became clear the need of new features that could automatically control each LED by receiving chords and notes as input.

4.3 Version 0.2

Still implemented in the same setup, this version added four modes to the instrument:

- The **song mode** is the main learning mode. A play along mode that displays the current chords on the fretboard.
- In the **scales mode** (improvisation mode), instead of chords, all possible notes included in the song's main scale were displayed. Therefore, the user

could improvise or practice solos while listening to the song even if they did not have previous knowledge of the scale's notes.

- The **animation mode** served as performance enhancement. The user could control changes in animations by pressing different buttons or the system could automatically change animations at a certain point in the performance.
- In the **"piano roll"** mode was a demo mode that displayed the notes of selected songs without the user's performance. The VioLED could play "Entre dos Águas" by Paco De Lucia without the musician.

4.3.1 Discussion

In this version, it became evident the main obstacles related to usability, such as anticipation of chords, occlusion and latency/jitter, as well as the need to test implementing the control software in mobile apps with wireless protocols. Furthermore, the overall memory usage of the embedded software became a concern, since it stored all the notes and chords.

Analyzing similar AMIS such as Fretlight, Fret Zealot and Populele, the main modes of operation are playing songs synchronized with the music and playing animations with the instrument. Differently from most guitar apps and AMIS such as Yousician, Guitar Pro, Fret Zealot and Fretlight, the VioLED uses a simpler chord chart instead of tablatures music notation, similar to the Populele and to music apps such as Chordify, FourChords.

Moreover, AMIS like Fret Zealot and the gtar have a distinct feature to show different colors for each finger, which shows musicians which fingers press specific notes in the fretboard. The VioLED uses a different approach showing the same color for every note, similar to the AMIS Fretlight and Populele. This considers that different colors may cause cognitive overload in the musician, since it is more information that needs to be processed every chord change. Another aspect is that different musicians may use different fingers to play the chords, choosing preferred hand positioning. However, future test is important to confirm if different colors can help or if it may cause cognitive overload.

The scales mode appears on some apps for the Fretlight, most serve as a music scale dictionary, but the Chords and Scales app also shows different scales for selected songs. However, by showing all possible notes in the scale, less experienced musicians may experience some of a "blank canvas" problem, not knowing how to begin or end musical phrases for improvisation.

4.4 Version 0.3

This version present changes in the control app. The system still utilized the core application from the previous version. However, the system used the TouchOSC iOS app as the UI (Figure 4). The TouchOSC app serves as a bridge to the Processing app, sending Open Sound Control (OSC) messages via WIFI protocol.



Figure 4: Song and Scale UI version 0.3.

4.4.1 Discussion

In this version, the control app only displayed which songs, animations and scales the user selected, without any music notation in the UI. The **song mode** implemented tempo change to help users that did not know the chord in the selected song. The user can reduce the song's speed to 80% of the initial BPM or increase it to 120%.

This version presented a more practical UI. The interaction with mobile devices reduces the time spent to start the task as well as the ease of use by controlling the system with a multitouch screen instead of randomly assigned keyboard keys as it was in previous versions. Moreover, it revealed the need to reduce the Arduino's memory usage, since all notes and chords for each song were in the Arduino by this version. That meant only a few songs could be implemented. Hence, the need for a new control app and a communications protocol.

4.5 Version 0.4

This version presented changes in the communications protocol, embedded app and control app. At first, the system implements a Bluetooth module (HC-06), with an Arduino microcontroller powered by a reduced mobile battery and controlled by an Android application.

Each song content was stored in JSON files over the internet and accessed by the control app. The instrument receives each chord wrapped in the communication protocol using Bluetooth (Figure 5).

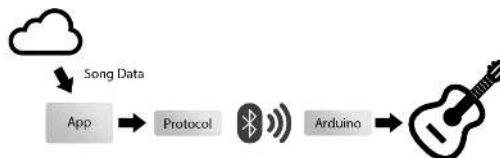


Figure 5: Version 0.4 scheme.

The communication protocol wrapper encapsulates each message sent to the VioLED in different bytes: start message, message type, message size, message (multiple bytes) and end message. Different messages can have different sizes depending on type of message (chords, metadata, color changes, etc.), chord notes, animations and others.

4.5.1 Discussion

At first, the chosen communication protocol was Bluetooth Low Energy (BLE). Despite the advantages of BLE's low latency, the 20-byte package size limitation was lacking to control animations. Since each LED uses 1 byte per color, the worst-case scenario needed 234 bytes, causing delay. Therefore, this version uses classic Bluetooth technology.

The app (Figure 6) still contains the same three modes: song mode, scales mode, animation mode. The user selects the song in a list and selects if he wants to play along (song mode) or improvise (scales mode). If the user selects song mode, the app shows the positions of the next chord as well as the current chord both in app and in the VioLED. Furthermore, the user can control the song's BPM to adjust the speed in the learning process.

Different from AMIS such as Fretlight and Fret Zealot, this version implemented chord anticipation in hardware. Before execution of each chord, an indication in red display each chord note before turning into green at the correct time (Figure 7). While most apps for the Fretlight do not anticipate chords, the Guitar Pro and the Fret Zealot relies on the anticipation of tablature notation on software. Without anticipation, the musician is surprised every chord change, since they may not know the next chord/note and its position in the fretboard. This also happens if anticipation is only in software, as the musician needs to change focus from the instrument to the app to check for changes or miss the anticipation.

The **scales mode** highlights the scale's key signature notes in red color to provide guidelines on possible beginning and end of musical phrases (Figure 8). Highlighting essential note in the scale can be a first step in attempting to solve the "blank canvas" problem.

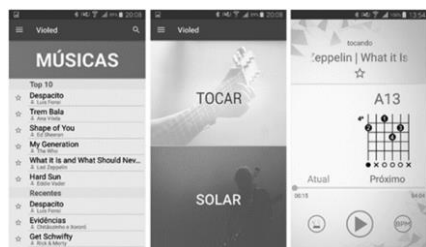


Figure 6: Version 0.4 Android app in song mode.



Figure 7: Chord anticipation in hardware.



Figure 8: Scale mode with highlighted notes.

This version still presents obstacles in usability regarding LED occlusion and the different possibilities of displaying content both in hardware and in the control app. The former is discussed in the next session. The latter relates to presenting musical information both in the instrument and in the app, which can cause cognitive overload and a bad learning experience. This suggests the need to perform tests with different versions of the system, using the app both as primary and as secondary "screen".

5. Preliminary results and discussion

The prototype's design process revealed a set of challenges and obstacles regarding different areas within the project. Most challenges involve usability aspects to display content and prototyping challenges when developing the hardware.

5.1 Obstacles in usability

Regarding content display, the identified challenges concerns time and the moment of execution of chords, occlusion of LED caused by the user's hands and feedback regarding the performance.

Preliminary results show that **chord anticipation** can help users to identify the position of next chords' notes on the fretboard. However, the system also needs to display indication of the exact moment of execution. However, fast chord/note changes may cause overwhelming sensations by constantly changing colors. Moreover, the amount of time to anticipate each chord and the means of anticipation may vary. Changing colors or blinking the notes can inform the user on the moment of execution. However, preliminary tests show that this anticipation needs to be at least on hardware, otherwise users less familiar with the song may be surprise by the notes appearing on the fretboard.

Occlusion is another obstacle in these systems. User's hand may hide additional notes or the anticipation of next notes and chords (Figure 9). This is a common problem in string instruments' AMIS. Anticipation of different notes and chords is difficult because the anticipation happens in the fretboard. However, Piano AMIS like the Yamaha Clavinova CSP [8] and the ones that use projection [10, 11, 12] solve this problem by not anticipating the notes on the actual piano keys. However, extending the visible area of the LEDs may help visibility, as in the gTar.



Figure 9: Occlusion cause by user's hand.

The instruments body can also occlude notes because the LED stay within the fretboard. In this case, several users may bend over the instrument to see clearly, which can compromise posture and eventually hurt users.

Possible solutions are reproducing the LEDs on the control app using computer vision, artificial intelligence, filming the fretboard and changing LEDs' position within the instrument. However, this requires further testing and hardware changes.

Another obstacle is providing reliable **feedback** by changing colors, brightness and providing audio cues from the software or from the hardware itself. As discussed in literature, complex tasks benefit from concurrent feedback, particularly in the learning phase, which can reduce cognitive overload [31]. Similar instruments and apps, i.e., Populele [7], Fret Zealot [6] and Yousician provide feedback over the user's device audio detection. The challenge is the feedback's reliability. Audio detection can compromise input and, therefore, compromise the feedback. More powerful artificial intelligence algorithms can help improve reliability at the possible cost in performance. One alternative is using sensors to detect the exact notes. However, this requires further testing since involves both hardware and software modifications.

5.2 Obstacles in hardware

When building these types of instruments, one must consider different **types of approaches**. Non-intrusive approaches may provide "sleeves" to place over the instrument, which facilitates maintenance and hardware modifications. However, the build needs to adapt to the instrument's dimensions and its limitations. As an example, when placing LEDs over the fretboard, one must consider the distances between each string, each fret and the space between the fretboard and the strings. Furthermore, placing the system over the fretboard can compromise overall ergonomics.

Contrastingly, intrusive approaches have limitations regarding hardware modifications, and system reproduction present the physical constraints of the "host" instrument. Component size and different instruments may affect the production of electronics components.

Another obstacle in hardware relates to **Latency and jitter**, which can lead to an increase in user error, affecting performance [32]. Therefore, latency also affects every choice regarding components, communication protocol and app development.

However, it is a subject fairly studied in the literature. Some authors provide guidelines on different usages and types of control [20, 28], discussing that highly demand systems may require latency below 10ms. When displaying notes and chords at execution time, the AMIS may require low latency or it could hinder synchronization, chord anticipation and, eventually, the overall performance.

Energy consumption also presents an obstacle in the development process. The selected RGB LEDs (WS2812B) operate in 5V and require 20mA per channel (60mA per LED on RGB). Therefore, the system requires approximately 5A to display every one of the 78 LEDs on white color.

In current version of the VioLED, the system applies a normalization of the color bytes. Generally, the

LED work with one byte per channel (0-255).

$$i_{system} = \sum_{n^o \text{ LEDs}} i_{LED}$$

Equation 1: Equations for overall system current and for each LED.

The normalization limits the sum of channel components to 120 instead of 765. The normalization reduces the overall consumption to approximately 16% of the worst-case scenario, leading to power supply reduction and system portability.

6. Conclusions and future research

This paper presented a study of Augmented Musical Instruments for Study (AMIS). The study shows that the design of these AMIS involves challenges regarding specific computer science areas as well as challenges music learning.

Using research through design method, a prototype is currently on development and the process behind developing its early versions already provided insights on usability and prototyping aspects. These relate to anticipation of chords, user feedback, latency and communication protocol requirements, minimum energy consumption, occlusion, etc. as well as generating hypotheses on possible solutions, which could guide future tests in similar instruments.

Chord anticipation and highlighting essential notes in music scales are important features in the learning process that current AMIS do not implement in hardware. On the other hand, occlusion still presents a, obstacle in the overall experience. Furthermore, the developed prototype still needs further improvement in feedback, as well as communications protocol and energy consumption optimization compared to AMIS in the market.

The development process corroborates with the challenges found in the literature. Each change in hardware/software affects the system as a whole and require further testing regarding latency, energy consumption, usability and tests with subjects to provide benefits to the learning experience.

Concerning future works, current versions of the control app, in development, implement features such as current chord duration, different formats to display content, and starts to integrate social network elements to share user's performance. The main concern is in how to display content between instrument and app in order to provide a better experience to the musician.

Results in preliminary user tests with 22 subjects in the Semana Nacional de Ciência e Tecnologia (SNCT) event suggest that the user's experience improved when learning new songs with the VioLED system when compared to traditional chord charts and the control app with a traditional acoustic guitar. The subjects noted that they felt "more confident" when using the VioLED. However, it is important to perform further testing with different groups and different setups to improve the system and produce design guidelines for similar systems.

References

- [1] Aumeo. (n.d.). <https://aumeoaudio.com/>.
- [2] E. R. Miranda and M. M. Wanderley. (2006). *New digital musical instruments: control and interaction beyond the keyboard* (Vol. 21). AR Editions, Inc.
- [3] L. Turchet, C. Fischione and M. Barthet. (2017). Towards the Internet of Musical Things.
- [4] K. L. Lueth. (2020). *State of the IoT 2020: 12 billion IoT connections, surpassing non-IoT for the first time*. IoT Analytics - Market Insights for the Internet of Things. <https://bit.ly/3yrD2ac>.
- [5] The Fretlight Guitar Store. (n.d.). *The Fretlight Wireless Guitar Store*. Retrieved August 6, 2021, from <https://fretlight.com/>
- [6] Fret Zealot. (2021). *Best Way To Learn Guitar | How To Learn Guitar At Home*. <https://www.fretzealot.com/>
- [7] Populele - Smart Ukulele (n.d.). *Populele Smart Ukulele by PopuBand Music Inc*. Retrieved August 6, 2021, from <https://popuband.com/>
- [8] Yamaha Clavinova CSP. (n.d.). *Clavinova CSP*. Retrieved August 6, 2021, from <https://yamaha.io/3yujPED>
- [9] S. Seol, Y. Shin and K. Lee. (2016). Learning Guitar with an Embedded System. *Contemporary Engineering Sciences*, 9(12), 553-560.
- [10] M. Löchtefeld, S. Gehring, R. Jung, and A. Krüger, guitar: Supporting Guitar Learning through Mobile Projection, CHI 2011 Extended Abstracts on Human Factors in Computing Systems, (2011), 1447-1452.
- [11] L. Raymaekers, J. Vermeulen, K. Luyten and K. Coninx (2014). *Game of tones: Learning to play songs on a piano using projected instructions and games*. In CHI'14 Extended Abstracts on Human Factors in Computing Systems (pp. 411-414).
- [12] Music Everywhere. *Music Everywhere - Instant Musician AR Piano for iPhone*. (n.d.). Retrieved August 6, 2021, from <https://www.music-everywhere.co/home>
- [13] T. Issa and P. Isaias (2015). Usability and human computer interaction (HCI). In *Sustainable design* (pp. 19-36). Springer, London.
- [14] Dix, A., Finlay, J. E., Abowd, G. D., & Beale, R. (2003). *Human-Computer Interaction* (3rd ed.). Pearson.
- [15] Nielsen, J. (2012). *Usability 101: Introduction to Usability*. Nielsen Norman Group. <https://bit.ly/3ysRyhQ>.
- [16] K. Y. Huang (2009, October). *Challenges in human-computer interaction design for mobile devices*. In Proceedings of the World Congress on Engineering and Computer Science (Vol. 1, No. 6, pp. 1-6).
- [17] G. De Michell and R. K. Gupta (1997). Hardware/software co-design. *Proceedings of the IEEE*, 85(3), 349-365.
- [18] D. Chen, J. Cong, S. Gurumani, W. M. Hwu, K. Rupnow and Z. Zhang (2016). *Platform choices and design demands for IoT platforms: cost, power, and performance tradeoffs*. IET Cyber-Physical Systems: Theory & Applications, 1(1), 70-77.
- [19] M. M. Wanderley and N. Orio (2002). *Evaluation of input devices for musical expression: Borrowing tools from hci*. Computer Music Journal, 26(3), 62-76.
- [20] S. Jordà (2005). *Digital Lutherie Crafting musical computers for new musics' performance and improvisation*. Department of Information and Communication Technologies.
- [21] C. Palmer and R. K. Meyer (2000). *Conceptual and motor learning in music performance*. Psychological Science, 11(1), 63-68.
- [22] N. MIKROS (2004) *Using Software Prototypes In Game Design*. In Game Design Workshop—Designing, Prototyping, and Playtesting Games. [s.l.] CMP Books, 2004. p. 460.
- [23] S. Oore (2005). *Learning advanced skills on new instruments*. In Proceedings of the 2005 conference on New interfaces for musical expression (pp. 60-64).
- [24] G. McPherson (Ed.). (2015). *The child as musician: A handbook of musical development*. Oxford University Press.
- [25] H. Pinto (1978). *Iniciação ao violão*. São Paulo: Ricordi, 1.
- [26] A. C. N. do Couto (2009). Música popular e aprendizagem: algumas considerações. *Opus*, 15(2), 89-104.
- [27] S. Jordà (2004, June). *Digital instruments and players: part I---efficiency and apprenticeship*. In Proceedings of the 2004 conference on New interfaces for musical expression (pp. 59-63).
- [28] D. Wessel and M. Wright (2002). *Problems and prospects for intimate musical control of computers*. Computer music journal, 26(3), 11-22.
- [29] J. Zimmerman, J. Forlizzi and S. Evenson (2007). *Research through design as a method for interaction design research in HCI*. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 493-502).
- [30] D. Fallman (2003). *Design-oriented human-computer interaction*. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 225-232).
- [31] R. Sigrist, G. Rauter, R. Riener and P. Wolf (2013). *Augmented visual, auditory, haptic, and multimodal feedback in motor learning: a review*. Psychonomic bulletin & review, 20(1), 21-53.
- [32] I. S. MacKenzie and C. Ware (1993). *Lag as a determinant of human performance in interactive systems*. In Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems (pp. 488-493).

CatalogToMakers: a collaborative cataloging platform for electronic components and physical computing projects related to musical creativity

Michael L. Bastos^{1*}, Felipe C. A. Calegario¹, Giordano R. E. Cabral¹

¹Centro de Informática – Universidade Federal de Pernambuco, Av. Jornalista Anibal Fernandes, s/n,
Cidade Universitária, CEP: 50.740-560 - Recife - PE - Brazil

mlb@cin.ufpe, fcac@cin.ufpe.br, grec@cin.ufpe.br.

Abstract. *The Maker Movement is a growing reality in the new industry and academic community. It consists of the Do-It-Yourself (DIY) style-spreading proposal, emphasizing the creation and development of creative thinking. In this context, the democratization of the Internet has meant that non-expert users are capable of developing new hardware technologies. In the music context, the easy access to low-cost equipment and the large number of people sharing knowledge facilitate prototyping and extend the creation of digital musical instruments. However, the places where the information related to the components used in these projects are not centralized nor organized. Therefore, it makes access to this information difficult, slowing down and hindering the flow of creative development. Thus, to reduce this development time and facilitate access to this information, this paper describes the processes for designing the CatalogToMakers tool, a collaborative cataloging platform for electronic components, and physical computing projects focusing on musical creativity. This approach implements several essential functions to facilitate the conception process of hardware development. Finally, an evaluation focused on non-expert users showed promising results.*

1. Introduction

Although it is an ancient activity, cataloging is commonly used today in different areas of knowledge. It aims to carry out two main tasks: the organization of information relating to a specific domain and providing and guaranteeing access to that information [1].

This method has been renewed continuously, especially after the advent of the Internet. The evolution of the web ended up making information increasingly accessible to a large part of the population. However, this constant and rapid dissemination of information ended up causing problems concerning the organization and retrieval of this information, making it difficult for the user to find the desired content of the more efficient way [2].

In order to support this, Information Retrieval (IR) appears as a procedure that searches for information in a specific document based on a set or collection. Currently, this information base can be understood as a repository or digital database [3]. In computing, those responsible for retrieving and displaying this information are the Informa-

tion Retrieval Systems (SRI), which perform the same process as IR, but in the digital environment, [2].

Given the diverse multidisciplinary areas in which SRIs operate, physical computing also becomes one of them. Physical computing is the process of transforming physical environments and objects through information processing using microcontrollers, actuators, and sensors [4], requiring in some cases the use of other components to support them, such as shields and batteries. Through the design and development of several instruments focused on new ways of sound expression and musical interaction [5].

In this context, the Maker Movement is currently a growing reality in the new industry and the academic community. It consists of the proposal to disseminate the Do-It-Yourself (DIY) approach, emphasizing the creation and development of creative thinking through physical computing and the development of new digital musical instruments [6].

Therefore, we hypothesize that obtaining information can be enhanced. This work aims to propose significant improvements in information retrieval systems related to electronic components and physical computing projects. Our approach is to design a collaborative cataloging tool.

2. Contextualization

The Maker Movement has been gaining proportion through the influence of teachers and enthusiasts who, by inserting the movement's proposal in traditional environments, encourage the creation and evolution of people's critical thinking. This concept is sufficiently disseminated in countries such as the United States and has increased interest in Brazil. In this new wave, the Fab Labs (Fabrication Laboratories) have been providing new aspirations, and philosophies for creating local ideas, managed by a global network based on this movement [7].

In this context, according to [8], Physical Computing is about the union of computing with electronics in the process of creating prototypes of physical objects through the use of microcontrollers, sensors, and actuators, allowing their interaction with human beings, having as main objective the connection of the physical and virtual worlds. In this way, ways of relating the use of computing in the interaction with other technologies are created, thus encouraging the development of solutions that can be used

*Supported by FACEPE.

in everyday life [8]. An area with the direct influence of physical computing is computer music, through the prototyping and development of new digital musical instruments and many other ways of music interaction and expression using sensors, actuators, microcontrollers, and many other types of electronic components [5].

Technological development with a focus on the process of organizing, searching, and retrieving information begins to gain importance in this environment because of the various difficulties in obtaining relevant information in the face of such large and unstructured spaces on the web [9]. In the context of prototyping in physical computing and computer music, authors [10] reports that there is a deficit in an environment with centralized information regarding the components used in designing new objects of innovation. These factors increase efforts and costs that could be reduced in a more organized scenario, with more specific functions and focused on users' primary needs [10].

In an attempt to assist in this process, some initiatives in the industry (Mouser.com, Filipeflop.com, Adafruit.com, Seedstudio.io, Labdegaragem.com, and Eletrogate, for example) and in the academy (such as Sci-fi Gesture Catalog and Sensor Wiki)[11][12] try to bring solutions that can help the creative process and improve access to information. However, it is possible to identify some gaps concerning usability, performance, and availability of information that, if filled, can bring good results about the learning of the target audience of these initiatives and the process of investigation and information retrieval of components and projects related to physical computing.

3. Development process

The design techniques used for the development of this project were based on the Design Thinking, developed by [13]. This method explores human-centered design. That is, it first explores the user's needs and then creates the product design. The phases of this process are divided into "inspiration," "implementation," and "evaluation".

These steps can be performed more than once in a circular fashion until a more solid and well-defined result is obtained. After developing these three steps, Design Thinking is structured within three crossed restrictions, they are:

- **Possibility:** it is directly related to what can be done;
- **Viability:** what can be practiced with the success of the project, and;
- **Desirability:** what the public wants or wants to see from the project.

Working based on these constraints, the designer begins to focus his efforts on highlighted problems and a structured project. This project is already started based on the specific problem and its probable solutions, using the identified objectives to solve them, based on the focus on the user.

For this work, the inspiration phase was carried out through the collection of pieces of information derived from the personal experiences of designers. The designers are part of the research group [blind review], in which several projects focused on Interaction Design have been developed, with the main focus on digital musical instruments. In the design stage, what was identified in the inspiration phase became solutions so that, in future phases, they could be improved. In this context, the methodology used was the same as that performed by [4] (Figure 1), where the author adapts the process proposed by Resnick [14], called Creative Thinking Spiral, which makes analogies to the children's learning flow in kindergarten, showing that they learn from the repetition of a cycle of activities.

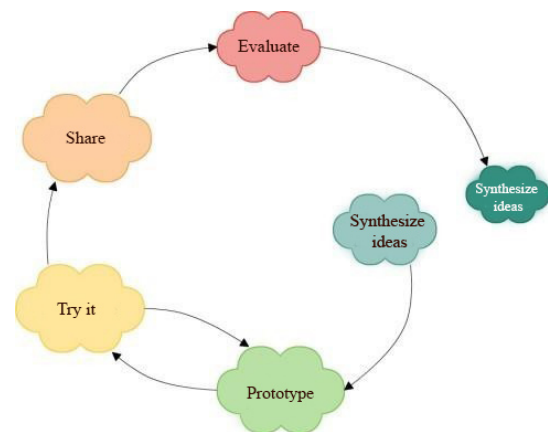


Figure 1: Spiral iterative process for the design phase

According to [4], using the spiral flow, the design stage developed for this approach is performed at least twice during the process, making it possible to improve the developed ideas. Thus, the main steps are described below:

- **Synthesize the ideas:** stage of organization of the collected data and summary of the key ideas;
- **Create prototypes:** step to create non-functional prototypes;
- **Try it:** stage where the low-fidelity prototype is executed with cycles for refinement;
- **Share:** test stage of the prototype with potential users;
- **Evaluate:** stage where the user's information is registered and the prototype is evaluated, if necessary, to start another phase of synthesis of ideas.

This entire process was developed based on the Maker audience, enthusiasts, and experts in physical and computer music, interaction design, and the development of connected objects using IoT.

Thus, after analyzing the methods and tools found in the literature, the ideas for the initial prototypes design phase were based on the most widespread methods of organizing and visualizing information in the last five years. This made it possible to build prototypes based on standards that were already widespread in academia and industry, focusing away from potential user's feedback on

technical issues and bringing them more into the functional modes of the platform. The first low-fidelity prototype was developed in the form of a drawing using storyboards (Figure 2). The screens were designed according to their sequence of execution, with each frame representative of the screen having a brief description of its functionalities.

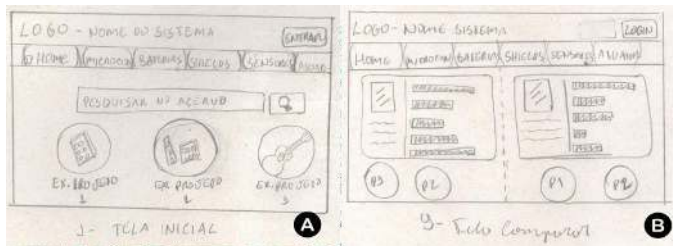


Figure 2: Example of screens from prototype 1 (A - Initial screen, B - Comparison screen between components).

Four potential users evaluated this first version, all experts in the field of computing or interaction design. Among these evaluations, questions were raised regarding new features and changes in the visual structure of some screens of the prototype. In this context, the information obtained was structured and synthesized to be inserted in the next version of the prototype.

After the ideation stage of the first prototype, the screens of version two were built, already enabling online access and navigation and identification of the flow between the screens. Figure 3 shows some of the screens created for the second prototype.



Figure 3: Some screens of prototype 2 (A - Initial screen, B - Comparison screen, C - Component management screen, D - Favorite screen)

The evaluation of the second prototype was carried out with a greater series of steps and with a more significant number of people (14). This time, two different profiles of evaluators were used, the specialists (also used in prototype 1) and potential technical users of the platform. The specialists work in computer science or interaction design, but who are not necessarily potential users, analyzing more the technical and visual part of the platform. On the other hand, the potential technical users are classified here as the people who continually use and need information regarding electronic components and or related

projects. Because they have this daily contact, they better understand the needs that a project of this type could supply.

Figure 4 demonstrates how the steps were carried out to evaluate this second prototype to understand better how this interaction with all the evaluators was carried out.

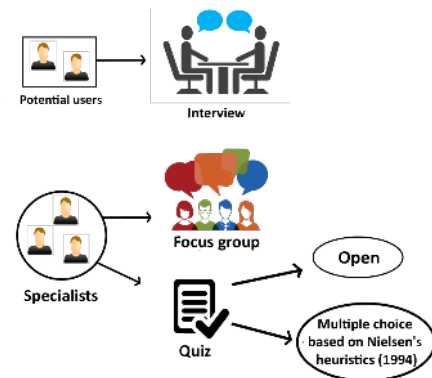


Figure 4: Mockup evaluation process (prototype 2)

The first assessment was made through a questionnaire applied to specialists, divided into multiple-choice questions, and an open field for recording possible suggestions for improvement. The multiple-choice questions were about usability and how information visualization happens on the platform, all of which were based on the Heuristics defined by [15], one of the best accepted and propagated heuristic evaluation methodologies in the literature for evaluating Human-Computer interaction.

4. CatalogToMakers

Based on the principles found in this study, mainly those that touch the methods used in the cataloging and organization of the information found through Literature Review, the CatalogToMakers was developed. The system appears as a tool to streamline the process of creating physical computing projects, whether related to musical computing, connected cities, smart homes, or similar, providing a collaborative cataloging tool, where the user can have centralized access to information in different ways, taking advantage of different features.

The platform allows the user to search, register, modify and delete components and compare components of the same category to understand more about which could be more suitable for a particular need. In addition, each component has related projects, with which the user can better understand the first steps of use, seeing in practice experiences developed by other designers. These related projects can also be registered by the tool's users, making possible a greater exchange of experiences among its users.

The great advantage of the tool concerning those already existing in the academy and the market is precisely the combination of several different functions. Each project can be evaluated and receive comments on its gen-

eral context. Working together allows a better user experience on a single platform.

Because it is a catalog, retrieving information must be quick and easy. For this, the user has three text search modes and a visual search mode. The textual search is done by item name, project name, and keyword, the latter being used for both projects and items. The keyword search field becomes important because it is the user who defines them (based on the principles related to Folksonomy), so the search can be done using non-technical terms registered by users, making it easier to find the item or desired project.

On the other hand, the visual search part is made through demonstrative icons of each category, that is, microcontrollers, sensors, shields, actuators, batteries, and projects. All of them will have a quick access button on the home screen, which means that even if the user does not know exactly what he wants to search for, he has access to the catalog items in general, exploring each item by the desired category. Figure 5 represents the flow of user interaction with the platform's search function. Potential users range from novice maker to the most technical and experienced.

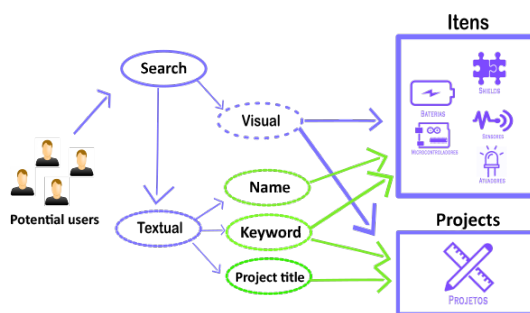


Figure 5: User interaction with the search function

These searches return a list with all the items or projects containing the text of their title or keywords, some sequence related to the characters typed. The initial screen of the application is shown in Figure 6.



Figure 6: Platform initial screen

The display of items related to visual search is done in card format, where each item or project appears with the highlight of its characteristic image and a brief description of its functions, in addition to a button that links

to the page related to it, Figure 7 shows an example of the listing of some items in the "Shields" category.



Figure 7: List of shields in the visual search

The result of the textual search is redirected to another page, where the components and projects are displayed in a list format. Each listed component has a favorites button (if the user wants to store that component in a list of his favorite items), a button with the link to view the component's information, containing its name, its average price and keywords referring to it, and another to perform a possible comparison with a component of its same category.

Figure 8 shows the component list screen after searching for the term "Arduino" in the search field name. This screen also has a quick search bar and a Home button, which takes the user to the home screen. Some buttons are displayed differently for users logged in to the system and for users not logged in at this point, and several others. Un-registered users do not have access to the favorites area or the item management screen.

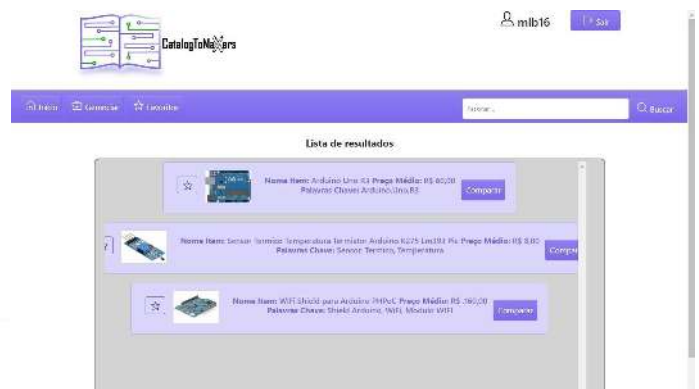


Figure 8: List of text search components

As seen above, one of the buttons that accompany the items is that of "compare." It is useful when the user wants to understand more about the technical possibilities of each component so that he can solve a specific problem, such as understanding which component consumes less battery or which wireless technology of a particular component is more suitable for some project.

In summary, the main functions of the application involve research of components and projects already registered (both in textual and visual pruning), comparison between components of the same category, management of

components and projects, favorite function, and a field for comments and evaluations of the users of the application for each project.

The entire development of the platform was based on the principles of Information Retrieval and the best cataloging practices currently reported in the literature, such as Folksonomy, Information Visualization (VI) principles, and some indexing methods. Regarding the implementation, technologies such as PHP5, CSS3, HTML5, Bootstrap 4, JavaScript, Ajax, and MySql were used as a database. All of this, structured in an MVC design pattern.

5. Implementation

The structure used in developing the final prototype for this project was based on the principles of Information Retrieval (IR) to provide the user with quick access to the catalog content. According to [2], IR is directly linked to the forms of representation, storage, search, and retrieval of information in information systems. This process mainly provides better communication between the user and what is indexed in the application.

Several versions of diagrams explore the information retrieval process, each one being developed based on the type of need for informational representation. Silva [2] demonstrates one of these diagram models (Figure 9) developed based on several studies of information representation models already carried out in the literature.

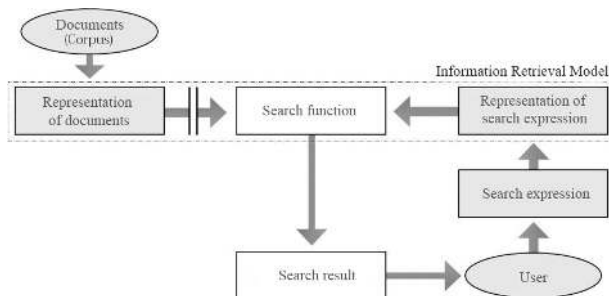


Figure 9: Representation of the information retrieval process

The CatalogToMakers was implemented to provide the best possible assistance to these steps used in the retrieval of information. That is why the phase of creating and evaluating low-fidelity prototypes was so important, serving to create a final prototype that was planned and focused on the user's needs.

As the system is a collaborative platform, that is, several people can insert and edit the catalog information as well as in a Wiki. The users also become catalog indexers since they will perform the process of representing the catalog document using its own terms. The platform also allows the insertion of Tags (keywords) to facilitate the information retrieval process further. Figure 10 demonstrates in a simplified way how the user indexes the information process, as well as how he has access to information from the catalog components.

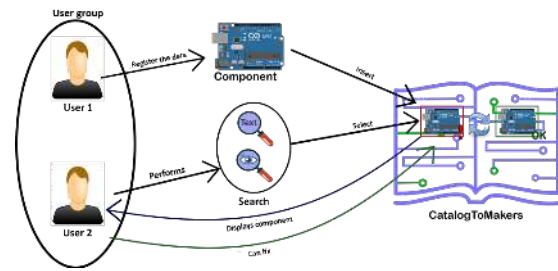


Figure 10: Catalog information retrieval and indexing process

In this example, user 1 performs the registration of information related to a specific component, where he himself chooses which terms best identify the element being registered. However, in this situation, one or more information is being registered incorrectly, going unnoticed by the indexing user who registered it. Then, user 2 performs a search for the same component and, through the search result, checks for incorrect information. At this point, the user edits the incorrect information and leaves the catalog updated again, and this modification can also be done to add new terms to the component, expanding the expressiveness of the element's representation.

6. Data Collection and Evaluation

The evaluation phase of the tool was divided into two stages. In the first, a face-to-face comparative analysis took place, and in the second, a usability analysis was carried out with a set of different potential users of the platform. In summary, this process was partitioned as follows:

- **Face-to-face comparative analysis:** in this process, the evaluators were invited to perform a series of tasks using the CatalogToMakers tool and Filipeflo e-commerce. The objective of this step was to analyze the user's reactions and the information retrieval process, as well as the time they were able to perform the tasks using both systems.
- **Usability analysis:** in this step, users were asked to use the platform (CatalogToMakers only) in an exploratory way, without a specific script, only with suggestions of where they could start. After that, the System Usability Scale [16] and an open questionnaire were applied to understand how the interactive process between the system and the end-user is being established.

6.1. Face-to-face comparative analysis

This stage of the analysis had the participation of 11 users, with the objective of understanding and comparing in practice the process of retrieving information from the CatalogToMakers tool and the Filipeflo website. It was done by executing a series of 10 Activities (5 the same for each system) by a group of users with different profiles and experiences, where each of these activities aimed to explore different functions in common to both systems.

All evaluations were filmed with the consent of the evaluators, with a view to identifying the time that each

performed the activities on the platforms and trying to find moments of difficulties and inconsistencies in the execution of tasks for operational or even visual reasons. In this context, participants should say whether or not they were able to perform a certain task within the flow of each application. At the end of the activities of both systems, the evaluators recorded their reports and suggestions regarding the experience they had with the platforms.

In carrying out all activities, CatalogToMakers demonstrated greater efficiency in the information retrieval process in relation to the Filipeflop site, being successfully executed by a more significant number of evaluators in all proposed tasks. The general list of the number of activities performed by the evaluators on each platform is seen in the graph in Figure 11.

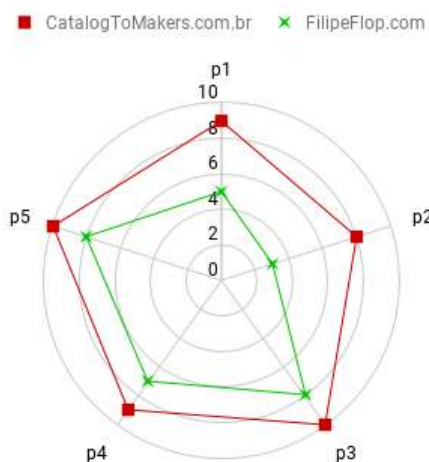


Figure 11: List of number of steps (P) performed by the evaluators for each platform

In addition, to analyze the agility in the access to information and the usability of the tools by the users, the time taken by each one to perform the five tasks proposed by the study on each platform was mapped. CatalogToMakers had its activities carried out in less time in 6 evaluations than 5 of the Filipeflop site. The relation of the time used by each participant in the execution of the tasks is shown below in Figure 12, where the evaluators were listed from A1 to A11 to better represent them.

For a better understanding of the difference between the times taken to perform the tasks in each application, a general average was also made between the times of all evaluators. In this survey, the differences in the execution period taken by the evaluators during their interactions with each application are more evident. This difference is reported in the graph in Figure 13.

The time that users take to perform the activities is directly linked to the information retrieval process. The sooner the user has the desired information, the more satisfied he will be. Applying it to creating a prototype, for example, ends up reducing the effort to find specific components for the project. It is also understood that this process can be improved, as there are still some inconsisten-

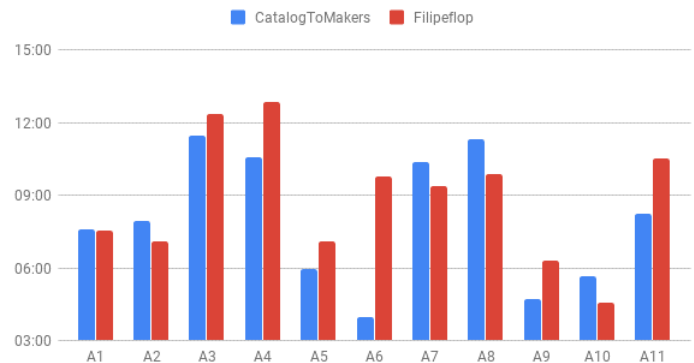


Figure 12: Time of execution of activities by each evaluator (A) on the CatalogToMakers and FilipeFlop platforms (Source: own author)

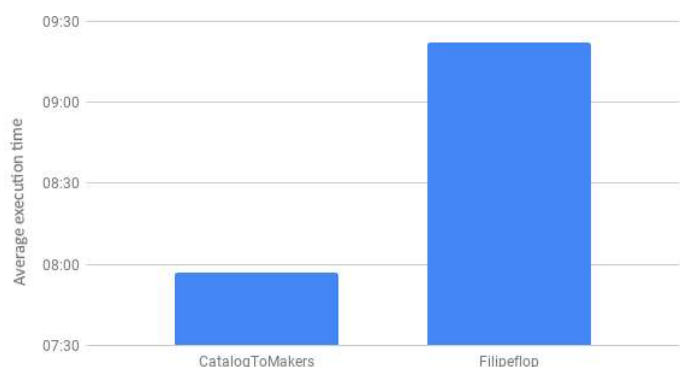


Figure 13: Average time of activities performed by the evaluators for the two applications

cies in the application that have been identified and can be improved in future iterations, further improving the time used to obtain this information.

7. Usability analysis

This stage of the evaluation process had the participation of 21 users, with the main objective of identifying information regarding the tool's usability process, as well as identifying opinions regarding the information retrieval process. The recruited users received instructions on how they could use the tool, and after exploration they were asked to answer a questionnaire, which was divided into three parts: User profile, System Usability Scale (SUS) [16] and Opinion survey, where the latter was divided into 5 open questions, which sought suggestions regarding the context of use of the tool, similar projects, profile of future users and negative and positive points of the same.

For this phase of the evaluation, 21 participants of different profiles were recruited, all Brazilian and with varying levels of knowledge. The affiliations were quite different, with components of graduate and undergraduate courses from different universities and technology institutes in the state of Pernambuco, where they are listed from P1 to P21 for a better organization of citations. Regarding the level of instruction of users on physical computing, the group remained well balanced, with similar percentages for each point, with the exception of level 1 (on a scale of 1 to 5) that had no members. Thus, 19% of the answers were from users who considered themselves with little knowledge about physical computing, choosing option number 2. Users with moderate knowledge, marked option 3, which corresponded to 28.6% of the total evaluators. Also with 28.6%, it was the users who considered themselves with a good level of knowledge about the area, choosing option number 4. And finally, with 23.8% it was the evaluators who marked option number 5, reporting having a lot of knowledge about the study area. This percentage relationship is reported in the graph shown in Figure 14.

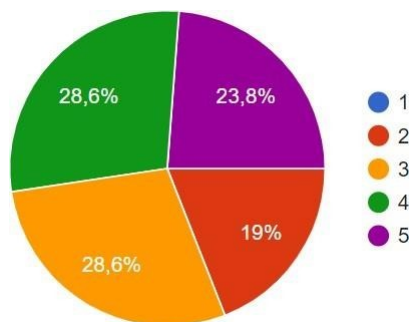


Figure 14: Level of knowledge of users about physical computing

After the process of identifying the characteristics of the evaluators, the SUS questionnaire and a set of 5 open questions were applied, in order to collect information about the context of use of the tool, possible user profiles for the application, related projects and the positive and negative points of the application.

The result of this application obtained a set of information that was analyzed and, in the end, the general SUS score was calculated based on what was defined by [?]. This information is detailed in Table 1.

Table 1: Result of applying the SUS questionnaire

Eval.	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
A1	5	1	5	1	4	1	4	1	4	1
A2	3	3	4	1	2	2	4	4	4	1
A3	5	2	4	3	5	1	5	1	4	3
A4	3	2	4	1	2	4	4	1	2	1
A5	3	2	3	1	3	4	5	1	5	1
A6	2	4	4	1	4	3	4	1	4	1
A7	5	2	4	1	4	2	4	1	4	2
A8	4	1	3	1	3	1	4	1	4	1
A9	4	1	5	1	4	2	5	1	5	1
A10	2	4	2	4	3	3	2	1	2	1
A11	3	4	5	2	4	2	5	1	4	2
A12	5	1	5	1	5	1	2	1	4	1
A13	5	2	3	4	4	2	4	2	4	1
A14	3	2	4	1	4	2	4	1	4	3
A15	4	4	5	1	5	1	5	1	4	2
A16	1	4	4	2	2	5	2	2	2	4
A17	4	1	5	1	3	4	3	1	5	1
A18	3	2	4	4	3	2	4	1	3	1
A19	5	1	5	1	5	1	5	1	5	4
A20	2	3	4	1	3	3	5	2	3	1
A21	5	1	5	1	5	1	5	1	4	1

Mean	3,61	2,23	4,14	1,61	3,66	2,23	4,04	1,28	3,80	1,61	Result (P1+P2+...P10) 2.5
(I) M-1 (P) 5-M	2,61	2,76	3,14	3,38	2,66	2,76	3,04	3,71	2,80	3,38	75,71

In the application of SUS, according to [17], the average score identified for web interfaces is 68.2. In this sense, the SUS score calculated for the CatalogToMakers based on the recorded evaluations, was higher than the average described in the study, resulting in a final score of 75.71.

8. Conclusions and Future Works

More broadly, the tool was successful in all evaluation criteria and, in particular, for less technical users (non-specialists). The main contribution of this investigation was the development of a solution for the information retrieval and organization process, enabling a more efficient search for components and projects used in physical and musical computing, having as a differential, in relation to the already existing platforms, the union of several important functionalities on a single platform and the use of essential techniques of design, computing and research information retrieval to promote a single experience to users.

For the development of future works, several recommendations obtained in the final validation process can be applied. Suggestions such as improvements in the interface, portability and the insertion of gamification to encourage the addition of information by employees, were considered of great importance to improve the project. In

this way, CatalogToMakers seeks to promote integration between enthusiasts, makers and more technical professionals, in view of the growth of this type of audience and the new demands inherent to society and industry in relation to these technologies.

References

- [1] M. J. M. Costa and V. P. da Conceição. *O Estado da Arte da produção científica em Catalogação nas regiões Norte e Nordeste*. RBBB. Revista Brasileira de Biblioteconomia e Documentação, v. 13 edition, 1996.
- [2] V. D. Vieira H. S. Silva and A. F. Lazzarin. *A Importância da Arquitetura da Informação para fins de Recuperação da Informação nas Perspectivas dos Sistemas de Navegação e Busca*. Revista de Biblioteconomia e Ciência da Informação, 2017.
- [3] F. R. Córdula C. M. Dantas and W. J. Araújo. *Análise Da Representação Da Informação Em Modelos Entidade Relacionamento Com Base Em Metadados*. Archeion Online, v. 4, n. 1 edition, 2016.
- [4] R. B Brasileiro. *tAMARINO: uma abordagem visual para prototipagem rápida em computação física*. Masters dissertation, 2013.
- [5] R. John. *DIY and Maker Communities in Electronic Music*. Cambridge University Press, 2017.
- [6] Y. P. T Sumitani F. R. Fowler and J. C. Noronha. *Hackathons como programas de desenvolvimento em empreendedorismo: um estudo de caso do Maker Hacklab*. 2016.
- [7] P. E. F. de Campos and H. J. d. S. Dia. *A insustentável neutralidade da tecnologia: o dilema do Movimento Maker e dos Fab Labs*. Liinc em Revista, v. 14 edition, 2018.
- [8] D. Z. Constantino. *Computação Física utilizando program-ME*. Assis, 2011.
- [9] D. M. Ambiner and Marcondes C. H. *As potencialidades da web semântica e web 2.0 para a ciência da informação e os novos formatos de publicações eletrônicas para a pesquisa acadêmico-científica*. EDICIC, v. 1, n. 4 edition, 2011.
- [10] M. Banz and M. Shiloh. *Primeiros Passos com o Arduino—2ª Edição: A plataforma de prototipagem eletrônica open source*. Novatec Editora, 2015.
- [11] L. S. Figueiredo. *A design method for building in-air gestural interactions*. 2017.
- [12] M. M. Wanderley. *SensorWiki. org: a collaborative resource for researchers and interface designers*. Proceedings of the 2006 conference on New interfaces for musical expression, ircam—centre pompidou edition, 2006.
- [13] T. Brown. with Katz, B.(2009) *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. 2009.
- [14] M. Resnick. *Sowing the Seeds for a More Creative Society. Learning and Leading with Technology*. s.l.: International Society for Technology in Education, available in; <https://web.media.mit.edu/~mres/papers/learning-leading.pdf>, accessed in: 06/19/2021. edition, 2007.
- [15] J. Nielsen. *A insustentável neutralidade da tecnologia: o dilema do Movimento Maker e dos Fab Labs*. ACM, 1994.
- [16] J. Brooke. *SUS-A quick and dirty usability scale*. Usability evaluation in industry, v. 189, n. 194 edition, 1996.
- [17] A. Bango, P. Kortum, and J. Miller. *Determining what individual SUS scores mean: Adding an adjective rating scale*. Journal of usability studies, 15th edition edition, 2009.

Hot Streaks in the Brazilian Music Market: A Comparison Between Physical and Digital Eras

Gabriel R. G. Barbosa¹, Bruna C. Melo¹, Gabriel P. Oliveira¹,
Mariana O. Silva¹, Danilo B. Seufitelli¹, Mirella M. Moro¹

¹Universidade Federal de Minas Gerais

Av. Antônio Carlos, 6627 – 31270-901 Belo Horizonte, MG

grgb@ufmg.br, {brunacamposmelo, gabrielpoliveira, mariana.santos, daniloboechat, mirella}@dcc.ufmg.br

Abstract. Consuming music through streams has made huge volumes of data available. We collect a part of such data and perform cross-era comparative analyses between physical and digital media for successful artists within the music market in Brazil. Given an artist's career, we focus on hot streak periods defined as high-impact bursts occurring in sequence. Specifically, we construct artists' success time series to detect and characterize hot streak periods for both physical and digital eras. Then, we assess their features, analyze them in the genre scale, and perform a cluster analysis to identify groups of artists with distinct success levels. For both physical and digital eras, we find the same clusters: Spike Hit Artists, Big Hit Artists, and Top Hit Artists. Our insights shed light on significant changes in the dynamics of the music industry over the years, by identifying the core of each era.

1. Introduction

Physical media is constantly making room for the consolidation of the digital era. Figure 1 shows this process in the music industry, with a turning point in the early 2010s. In such a market, physical media sales are still going on; whereas streaming services dominate music consumption, accounting for over 62% of the music industry revenue in 2020.¹ The scenario in Brazil is similar. As Latin America's largest music market, about 72.4% of music revenue comes from digital media in Brazil, against 1.4% from physical, according to the most recent Pró-Música report.²

The current streaming dominance has proved beneficial in several ways, such as promoting local artists and increasing listeners' engagement. Also, while the world contends with the COVID-19 pandemic, people have once again been reminded of the timeless power of music to console, heal and lift their spirits, which strengthens the connection between artists and fans. Indeed, with the world in lockdown and live music shut down, most fans around the globe enjoy music via streaming. Still, streaming popularization has also brought new challenges due to the massive volume of music-related data to process and analyze.

An example of a task that has become more complex and important in recent years is finding and promoting artists with promising careers. If a contract with a major label was essential to be successful in the physical era, artists from smaller or independent labels can now go viral and become popular thanks to streaming services. Thus,

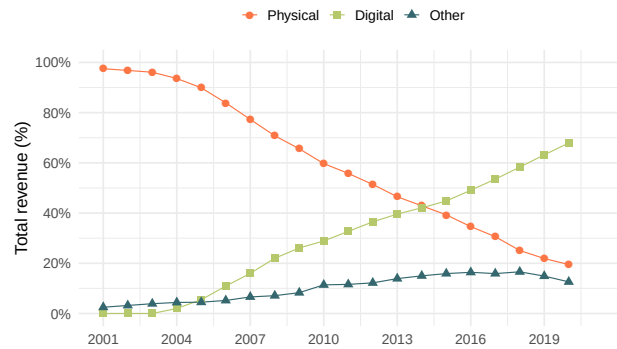


Figure 1: Global recorded music revenues by segment (2001–2020) – source: IFPI GMR¹

Artificial Intelligence (AI) tools and techniques come to hand and assist in this type of analysis, creating significant benefits to both the artists and the A&R (artists & repertoire) executives. In fact, many applications use AI-powered technology in the music industry, such as genre classification [1, 2] and success prediction [3, 4]. Regarding the latter, identifying upcoming artists with outstanding success is crucial, as it helps planning and adjusting marketing directions for their careers.

Overall, musical careers usually present continuous periods of success above average, defined as *hot streaks* (HS). This concept has been explored in several domains, including science [5], social media [6], and creative careers [7, 8]. In such a context, our goal is to identify and characterize hot streak periods in the music market in Brazil. We assess the evolution of successful careers by comparing data from physical (1990–2015) and digital (2016–2020) eras. In particular, we build artists' success time series based on sales (physical era) and streams (digital era) to detect hot streak periods. Then, we perform a cluster analysis to group artists according to their success level. Finally, we characterize such hot streaks to extract insights about temporal evolution of musical careers.

The remainder of this paper is organized as follows. First, we discuss related work in Section 2. Then, we describe the data acquisition process in Section 3. We detail the methodology used to identify hot streaks in Section 4. Next, we apply such methodology in the physical and digital eras in Sections 5 and 6, respectively, with a cross-era comparison in Section 7. Finally, we conclude with future directions in Section 8.

¹IFPI Global Music Report: <https://gmr2021.ifpi.org/>

²Pró-Música Brasil Report: <http://bit.ly/ProMusica2019>

2. Related Work

Although streaming platforms are inherently designed to not interfere with the music production process, their leading role in the music industry is unquestionable: they determine the amount paid to music content producers, and dictate the type of music accessible through their recommendation algorithms [9]. Such constant changes in the music market reinforce the investigation demand on the implications of these new media players' insertion. Specifically, the physical to digital era shift requires attention mainly for Music Information Retrieval (MIR), not only on technological-driven factors [9, 10, 11] but also on the success patterns shaping this dynamic market [4, 12, 13].

After decades of intense transformations in the music market, the digital era brought novel challenges, including a substantial volume of data. As human inspection is almost impossible for music big data scale, specialized algorithms can help with several tasks in MIR, including music recommendation [14], automatic genre classification [1, 2, 15], and so on. Another possible benefit is to feed machine-learning models for musical success early prediction, contributing to identify trends and new talent. Indeed, evaluating the impact of human performance is a common practice in many research fields [16, 17, 18]. The term *hot streak* emerges in such context, as the reference to a specific period within professional careers when the success is significantly higher than the average [7].

For individual and creative careers, research assessing impact is much more recent. Liu et al. [7] consider large-scale careers of artists, film directors, and scientists to demonstrate that hot streaks are remarkably universal across diverse domains, yet usually unique across different careers. In this sense, Garimella and West use data from Twitter, one of the most popular online social networks, and define users' impact as the reach of their content [6]. Janosov et al. [8] also consider luck as a crucial ingredient to achieve impact in creative domains. Regarding music, they model the historical artist timelines based on the release year of songs and measure success by the total play counts obtained from Last.fm.

Nonetheless, to the best of our knowledge, no previous studies address the dynamics of music artists' success periods (i.e., hot streaks) within the Brazilian market. Although Brazil's high rates of music consumption, little is known about the key factors driving musical success and defining artists' promising careers. As regional markets have their own success patterns and behavior [13, 19], such individual analyses are crucial. Therefore, this work is a step forward towards understanding the specific dynamics of music artist success within the Brazilian market.

3. Data Acquisition

To perform a cross-era comparative analysis between physical and digital media, we focus on musical success in Brazil. Our first data source is Spotify, the most popular global audio streaming service. However, its Charts only comprise data from 2016 onwards. Hence, to describe the

Table 1: Pro-Música Brasil certification levels for Brazilian and foreign artists (A&S is Albums and Singles).

Certification	Brazilian		Foreign	
	A&S	DVDs	A&S	DVDs
Gold	40,000	25,000	20,000	15,000
Platinum	80,000	50,000	40,000	30,000
Double Platinum	160,000	100,000	80,000	60,000
Triple Platinum	240,000	150,000	120,000	90,000
Diamond	300,000	250,000	160,000	125,000
Double Diamond	600,000	500,000	320,000	250,000
Triple Diamond	900,000	750,000	480,000	375,000
Quadruple Diamond	-	1,000,000	-	500,000
Quintuple Diamond	-	1,250,000	-	625,000

Digital Era, we consider the range period available (2016–2020). Then, we also use the Pró-Música Brasil platform to describe the Physical Era, with data from 1990 to 2015. Next, we detail data acquisition processes for both physical (Section 3.1) and digital media (Section 3.2).³

3.1. Physical Media

Pró-Música Brasil (PMB) is the official representative body of the record labels in the Brazilian phonographic market. It represents artists in legal and financial instances and issues certification awards, as authorized by record companies. The certification awards recognize the work of performers according to sale numbers in the form of “special discs”, i.e., Gold, Platinum and Diamond discs. The data on such awards is available on its website⁴ and was collected on February 5th, 2021. The final dataset comprises information on awarded artists, release year, disc category, song/album name and media type since 1990.

In PMB, the threshold sales number for each certificate depends on whether the artist is Brazilian or not, as shown in Table 1. However, as such information is not available in PMB, we crawled it from Wikipedia using a Python library.⁵ Next, we collect the total sales for each musical work based on the certification awarded, its nationality and PMB's sales metric for the disc award. Finally, we use Spotify's API⁶ to associate each artist to their respective genres specified on the streaming platform. Hence, our final dataset contains information about 4,198 musical works from 780 artists. Considering only the period between 1990 and 2015 (i.e., Physical Era), there are 3,243 musical works from 574 artists. Quantitative information on the certificates is shown in Figure 2 (left).

3.2. Digital Media

Between 2016 and 2017, there was a key change in PMB's metric, which moved from physical media (i.e., DVD and CD) towards digital media (i.e., Singles and Albums), as depicted in Figure 2 (right). Meanwhile, streaming was already the main revenue source for digital media (58.3%).⁷

³Link to whole datasets omitted for blind review

⁴PMB Certificates: <https://bit.ly/CertificatesPMB>

⁵Wikipedia Python Library: <https://github.com/goldsmith/Wikipedia>

⁶Spotify API: <https://developer.spotify.com/>

⁷Pró-Música Brasil 2016: <https://bit.ly/ProMusica2016>

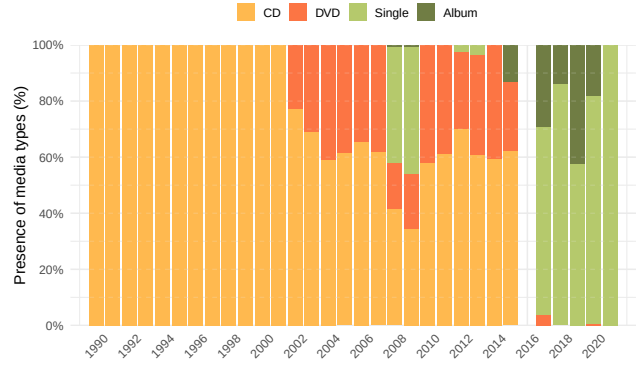
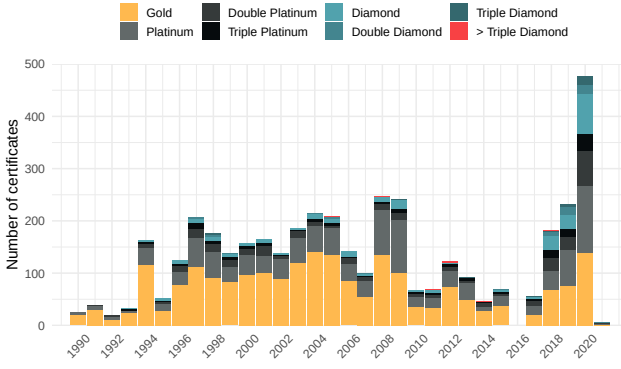


Figure 2: Discs certificated (left) and media type (right) in Pró-Música Brasil (1990–2021). In 2016, there was a metric change in the certification, hence the lack of data.

Given its relevance, we extract data referring to the Digital Era from the weekly Spotify Top 200 Chart, which corresponds to the most streamed songs in Brazil. Each chart entry contains the song’s name and its artist(s), the number of streams, the song’s Spotify URL and its position on the chart. We collect data from January 2017 to December 2020. We also collect artist data using the Spotify API: name, number of followers, and their list of genres. Our final dataset for Digital Era comprises 2,595 songs from 1,018 artists obtained from 108 weekly charts.

4. Hot Streak Detection

After collecting the data, we are able to build artists’ success time series for both eras (details in Sections 5 and 6). To detect hot streaks in such time series, we rely on previous work that shows the most successful points in professional careers tend to happen close to each other [6, 7]. Hence, we use a technique to reduce the time series dimensionality to delimit continuous periods within careers. Then, we define a hot streak as the periods in which the success (i.e., physical sales or digital streams) is above a certain threshold obtained from the career itself. In other words, the hot streak detection does not consider external factors (e.g., genre and time) because artists reach different levels of success, and choosing a single threshold would make the comparison unfair.

To reduce time series dimensionality, we use Piecewise Aggregate Approximation (PAA) [20]. Given a time series $X = x_1, x_2, \dots, x_n$ of length n , PAA reduces it into a new series $\bar{X} = \bar{x}_1, \bar{x}_2, \dots, \bar{x}_N$ with N dimensions, $1 \leq N \leq n$. The intuition behind this method is that dividing the original time series into N equal-sized segments produces N new points. The value of each segment is defined as the average of the points within such a frame (Equation 1). Hence, the approximation of each point on the original time series is made by simply assigning the PAA value of its corresponding segment.

$$\bar{x}_i = \frac{n}{N} \sum_{j=\frac{n}{N}(i-1)+1}^{\frac{n}{N}i} x_j \quad (1)$$

Note that artists’ careers may contain points with extreme values for the success metric. Therefore, PAA is

a helpful tool to smooth such differences and delimit periods in the careers. Regarding code, we use the PAA implementation of *tslearn* [21], a Python package for time series analysis. Its only parameter is the number of segments to split the series into (further information on values next).

Finally, for each artist, we chose a specific threshold for defining the hot streak periods. Such an individualized approach is based on the percentiles of the success metric, and it allows analyzing careers of artists with different levels of success. In other words, as success is relative for each artist, we detect HS for widely known artists with higher sales and streams, as well as independent artists who received only a few certificates and streams.

5. Physical Era

Here, we deepen the analysis on the PMB dataset by building the time series for each artist in Section 5.1, and characterizing the hot streaks and understanding their relationship to music genres in Section 5.2. Furthermore, we cluster artists into different success groups in Section 5.3.

5.1. Artists’ Time Series

In the Physical Era, the evolution of an artist’s success is represented by the certificates received at the PMB from 1990 to 2015. We build the time series with an annual granularity because PMB provides data for the year the artist got the certificate. Hence, each point in the artists’ time series corresponds to the number of sales achieved by the artist in that year. This number is equivalent to the aggregate of certificates received (Table 1).

From the artists’ time series, we detect the hot streaks periods using PAA (see Section 4). To do so, we first set the number of segments in which the series will be split, as this is the only parameter of the method. Since we deal with yearly success time series, the minimum size of each segment must be two years. After extensive empirical experiments, we set the window to the minimum, which is enough to validate a hot streak. Hence, the number of segments is calculated by dividing the time series length by such a predefined size. In addition, we set the 80th percentile of the success metric in artists’ time series as the threshold for defining the hot streak periods.

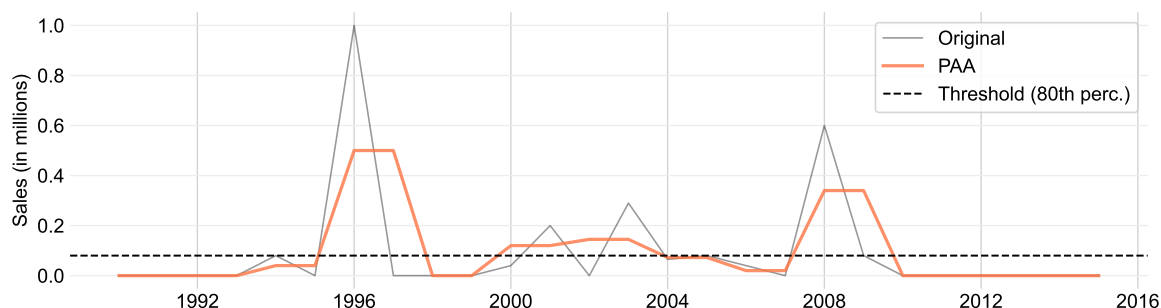


Figure 3: Piecewise Aggregate Approximation (PAA) applied to Skank's success time series in the Physical Era (1990–2015). Periods above the threshold are considered hot streaks.

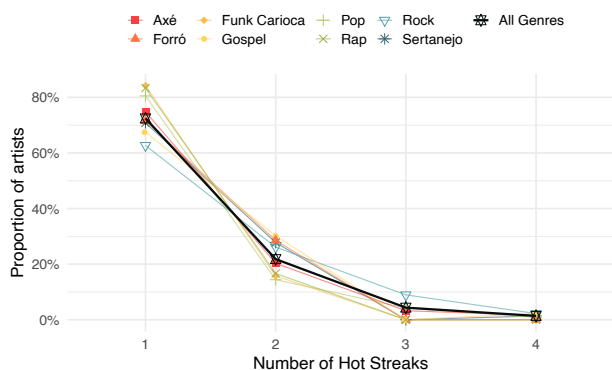


Figure 4: Hot streaks detected for artists of distinct genres in the Physical Era.

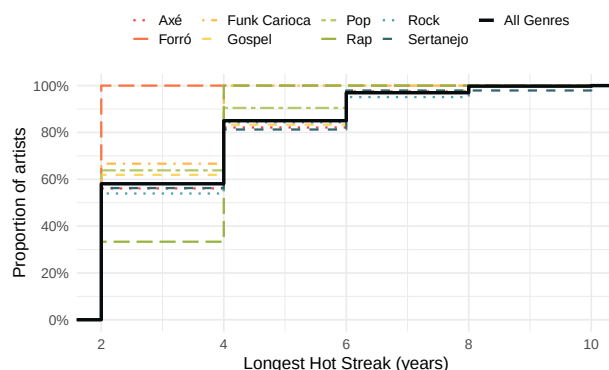


Figure 5: Cumulative distribution function of the duration of the longest hot streak in the Physical Era (in years).

In order to showcase how the hot streak detection works, we use the time series of the Brazilian band Skank. Figure 3 shows their time series after applying PAA. The threshold is set as the 80th percentile, as aforementioned. We observe three distinct hot streak periods. The first one from 1996 to 1997, when they released *O Samba Poconé*, a diamond disc awarded album. The second hot streak lasted from 2000 to 2003, when they released the albums *Maquinarama* and *Cosmotron*, which granted them platinum and gold discs, respectively. The last one occurred from 2008 to 2009, when they released the album *Estandarte*. Skank is the perfect example of a successful album that does not lead to a HS: in 1994, their album *Calango* got two golden discs, which is still inferior to their greatest hit *Garota Nacional* released in 1996.

5.2. Hot Streak Characterization

We characterize the hot streak periods identified for artists in the Physical Era according to their musical genres. As individually considering closely related music styles would create an artist overlapping and bias the results, we create super-genres for this analysis. For example, we verified that *Indie Folk* was more associated with *Rock* than any other super-genre and was then incorporated to *Rock*. Finally, the top eight prominent super-genres considered to the Physical Era in Brazilian music are *Pop*, *Rock*, *Sertanejo*, *Rap*, *Axé*, *Funk Carioca*, *Gospel* and *Forró*.

Figures 4 and 5 assess each genre's performance regarding the number of hot streaks as well as their maximum duration, respectively. In Figure 4, the majority of

artists (about 70% to 94%) have up to two hot streak periods, and only 6% of the artists have between three and four hot streaks. Although such a pattern happens in all considered genres, there is no clear correlation between genre and the number of HS. As we consider a yearly time window, achieving a high number of hot streaks is not an easy task.

Next, we analyze the duration of hot streaks. We consider only the longest HS for each artist, as they may experience more than one. In Figure 5, we use a Cumulative Distribution Function (CDF), which informs the proportion of artists who have the longest HS with duration up to that value for a given number of years (x-axis). Note that the size of a HS is always be a multiple of two, as the PAA segment is set to two years (see Section 5.1). In general, only *Pop*, *Gospel* and *Funk Carioca* genres follow a similar behavior when compared to the average (i.e., *All Genres*). While the *Rap* genre has artists with shorter HS (up to four years), artists from the other genres have a wider range (between one and ten years). Therefore, in contrast to the previous analysis, we note that genre is relevant to describe the longevity of hot streaks.

5.3. Cluster Analysis

We now move to the cluster analysis, which helps better understand the characteristics of the different success levels of artists achieved during the Physical Era. We use the K-Means algorithm, which is the most commonly used clustering method for dividing a dataset into a set of k groups. The considered features for the algorithm include

Table 2: Main statistics on the artist clusters in the Physical Era.

	All	SHA	BHA	THA
Number of Artists	574	527	38	9
Average Number of HS	1.3	1.3	1.8	1.9
Median Sales (10^4)	8.5	8	152	507
Median Threshold	0	0	72,500	300,000

the total number of hot streaks, total sales and the time series threshold. In order to determine the optimal number of clusters, we use the Elbow Method [22] that plots the explained variation as a function of the number of clusters and consider the curve *elbow* as the optimal k . In our case, the method outcome suggests $k = 3$. We name the resulting clusters according to the success metric (i.e., number of sales): *Spike Hit Artists* (SHA), *Big Hit Artists* (BHA), and *Top Hit Artists* (THA). The main statistics of the clusters are presented in Table 2 and summarized as follows.

Spike Hit Artists (SHA). This cluster contains most artists (527). The median PAA threshold for such artists is 0, indicating that their sales are, in general, much lower than the artists from other clusters. In addition, the average number of hot streaks for SHA is 1.3. Such a result suggests that PMB certificates happen sparsely for such artists because there are not many hot streaks even with a low threshold. Artists in this cluster include Banda Eva, Claudinho e Bochecha, Coldplay, Paramore, and Latino.

Big Hit Artists (BHA). This cluster represents a bridge between the most and less successful ones, with 38 artists. Although BHA presents a higher number of sales when compared to SHA, they do not present a proportional increase in the average number of hot streaks. In fact, they have on average 1.8 hot streaks between 1990 and 2015. Besides, there is a substantial increase in the threshold for artists in this cluster, reaching 72,500 sales. Madonna, Legião Urbana, Skank, and U2 are examples of BHA.

Top Hit Artists (THA). This is the cluster with the most successful artists, as they have the highest median number of sales. The average number of hot streaks for THA is very close to the value for BHA. However, the artists in this cluster achieved major sales success throughout the Physical Era, as they have a very high median threshold (300,000 sales). Examples of THA include Ivete Sangalo, Zeca Pagodinho, Roberto Carlos, and Sandy e Junior.

Figure 6 presents a scatter plot comparing the number of hot streaks and the total number of sales per artist. The results confirm the artists are clustered according to success level, i.e., total number of sales. Regardless of their number of hot streaks, SHA generally present the lowest, followed by BHA and THA.

6. Digital Era

We now deepen the analysis on the Digital Era by building the artists' time series that achieve Spotify's Brazil Top 200 Charts in Section 6.1. Next, from the constructed time series, we detect and characterize hot streaks in Section

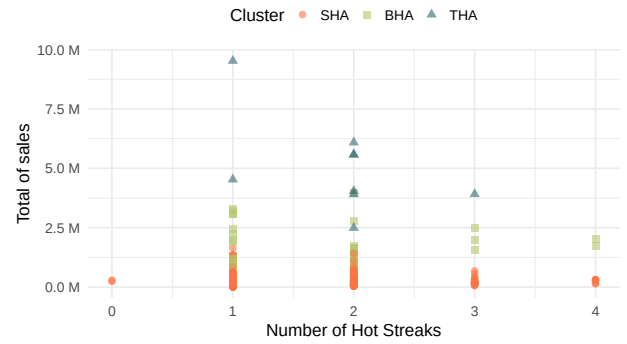


Figure 6: Number of hot streaks and total sales for clusters SHA, BHA, and THA.

6.2. Finally, we perform a cluster analysis to group similar artists based on their success levels in Section 6.3.

6.1. Artists' Time Series

In the Digital Era, we use the Spotify Top 200 Chart in Brazil as our basis to model artists' success over time. For each artist, each point in their time series represents the accumulated success in a given week, according to the chart. In this case, the success measure considered is the total number of streams (i.e., the number of times the song was listened to on the Spotify) per week. For example, Figure 7 presents the time series of the Brazilian singer Anitta.

Similar to the Physical Era, we apply PAA (Section 4) in each time series to detect the hot streaks periods. This method helps balance weeks with little or no success metric value, which was quite common in the data. In the Digital Era, songs and artists achieve success much faster. In addition, the digital nature of streaming platforms allows for successful weekly data to be made available almost in real-time. Hence, according to our experiments, we define 12 weeks as the size of each PAA segment. Each segment comprises a three-month period, which is a reasonable time to analyze the continuous periods of great success in streaming platforms such as Spotify.

We define hot streaks as the periods in which PAA is above a threshold, defined per each artist. However, as we are dealing with numerous single weeks on the Spotify Chart, we consider the artists' Activity Rate (AR) based on the threshold. The AR is the number of weeks that an artist appears on the chart divided by the total number of weeks. Hence, if $AR < 10\%$, the threshold is set as 95% of the success measure; if $10\% \leq AR < 15\%$, the threshold is set as 90%; if $15\% \leq AR < 20\%$, the threshold is set as 85%; finally, if $AR \geq 20\%$, the threshold is set as 80%.

Figure 7 depicts PAA applied to Anitta's career, currently one of the most influential Brazilian artists worldwide. She is present in all considered weekly charts ($AR = 100\%$) and thus, we set the 80th percentile as the threshold for defining hot streaks. There are three HS in her time series. The first one from November 2017 to April 2018, when she released *Vai Malandra*, which became the most streamed song on its release date.⁸ The second HS

⁸No Brasil, "Vai Malandra" supera Taylor Swift e é a música

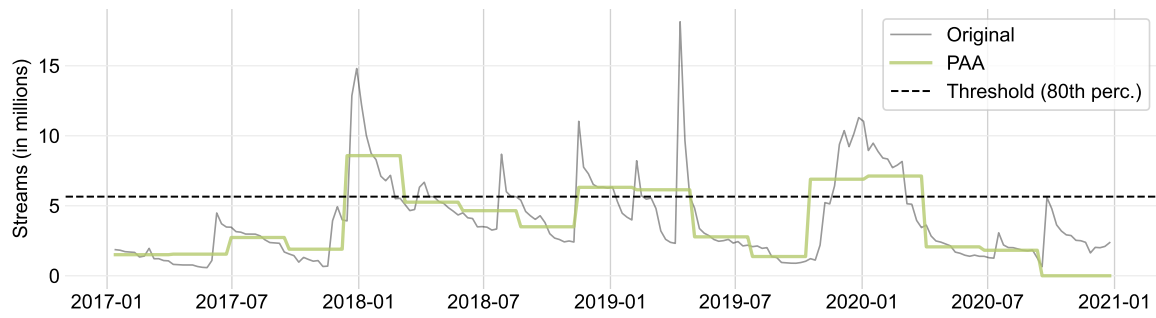


Figure 7: Piecewise Aggregate Approximation (PAA) applied to Anitta's success time series in the Digital Era (2017–2020). Periods above the threshold are considered hot streaks.

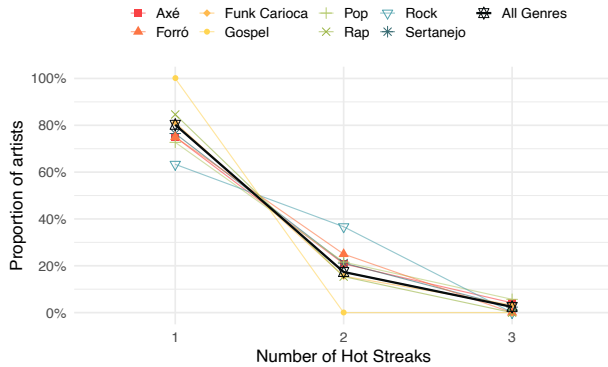


Figure 8: Hot streaks detected for artists of distinct genres in the Digital Era.

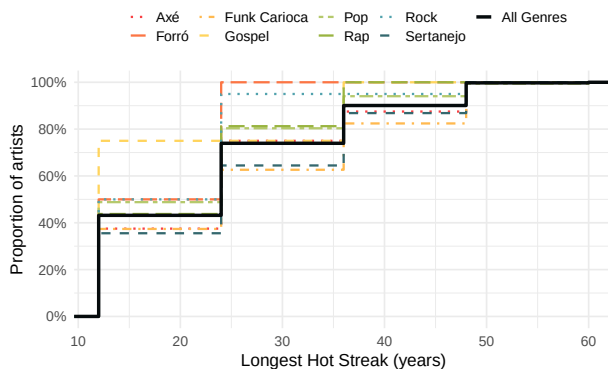


Figure 9: Cumulative distribution function of the duration of the longest hot streak in the Digital Era (in weeks).

period coincides with *Veneno* and *Não Perco Meu Tempo* single releases. Finally, the third one comprises the period in which she collaborated with both national and international famous artists, such as Marília Mendonça (*Some Que Ele Vem Atrás*) and Black Eyed Peas (*Explosion*).

6.2. Hot Streak Characterization

We characterize the hot streak periods identified in the previous section. As in the Physical Era, we also consider the same super-genres in our analyses. Figures 8 and 9 present the HS analysis in quantity and duration perspectives, respectively. Regarding the number of hot streaks,

mais tocada no Spotify em sua estreia: <https://bit.ly/AnittaVaiMalandra>. Accessed on June 26, 2021.

Table 3: Main statistics on the artist clusters in the Digital Era.

	All	SHA	BHA	THA
Number of Artists	1,018	940	70	8
Average Number of HS	0.6	0.5	1.6	1.6
Median Streams (10^5)	7.6	3.9	2,190	10,857
Median Threshold	0	0	2,052,131	7,145,116

most artists (about 80% to 98%) have up to two hot streak periods, similar to the Physical Era. Besides, only about 2% of artists achieved three hot streaks in the Digital Era. Although all genres follow a similar trend in general, *Rock* presents a higher percentage of artists with two HS. In contrast, all *Gospel* artists have only one HS in such period.

Next, we assess the duration of hot streaks. Analogous to the Physical Era, we consider only the longest HS for each artist. Overall, about half of the artists from all genres have three-month long hot streaks. However, when analyzing the genres individually, there are specific hot streak patterns. For instance, most *Axé* artists have shorter HS periods, as more than 60% of them present a 12-week hot streak. On the other hand, *Sertanejo* artists have longer hot streaks, as about 60% of them have HS periods with up to 24 weeks, i.e., six months.

6.3. Cluster Analysis

We now perform cluster analysis over the Digital. We also apply the K-Means algorithm in the time series, and the Elbow method to find its optimal number of clusters. As the Physical Era, the optimal k number found is three. We name the resulting clusters according to the success metric (i.e., number of streams); as the results are similar to the Physical Era, the names remain the same. The clusters main statistics are in Table 3 and summarized as follows.

Spike Hit Artists (SHA). This cluster comprises most artists (940). The median PAA threshold for such artists is 0, indicating that their streams are, in general, lower than other clusters. In addition, the average number of hot streaks for SHA is 0.5. However, we cannot say that such artists were not successful because they are in Spotify's Top Charts. On the other hand, they do not have many hot streaks (one HS on average). Artists in this cluster include Billie Eilish, Leo Santana, and Naiara Azevedo.

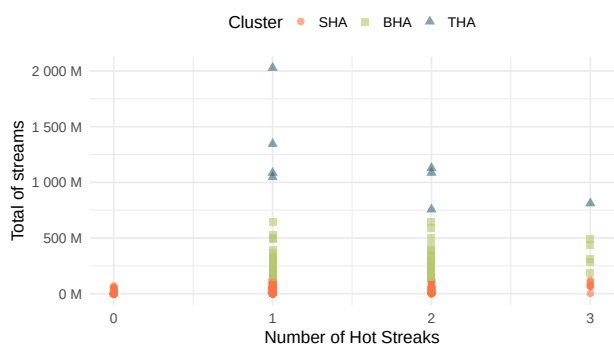


Figure 10: Number of hot streaks and streams for clusters SHA, BHA, and THA.

Big Hit Artists (BHA). This group represents a bridge between the most and less successful ones, comprising 70 artists with an average of 1.6 hot streaks between 2016 and 2020. Besides, the median streams are considerably higher compared to the previous group, as is the threshold, reaching more than 2M streams. Barões da Pisadinha, Dua Lipa, and Pablo Vittar are examples of BHA.

Top Hit Artists (THA). This cluster contains only eight artists, with an average of 1.6 hot streak periods per artist. The number of Spotify streams is much higher when compared to the previous groups, which is also observed with the threshold. Therefore, Top Hit Artists may be considered highly successful artists, as their songs achieve a higher stream count throughout the weeks. All THA in the Digital Era are Brazilian, and examples include Anitta, Marília Mendonça, and Zé Neto & Cristiano.

Figure 10 presents a scatter plot comparing the number of hot streaks and the total number of streams for each artist. The results confirm that the artists are clustered according to their success level. Overall, all clusters patterns are similar to the Physical Era. Specifically, SHA generally present lower sales, while THA are the ones with higher sales. Furthermore, BHA have an intermediate level of sales. Hence, our clustering approach is coherent, as it manages to distinguish the levels of successful artists.

7. Cross-era Comparison

Music is part of people's daily lives regardless of the era experienced, whether Physical or Digital. With musical consumption constantly on the rise, we may notice similarities between both eras. From the results obtained in Sections 5 and 6, we explored Hot Streaks (HS) in musical careers within the Brazilian market. Such HS periods provide valuable information used in cluster analysis, in which we also notice cross-era similarities. We highlight the BHA and THA clusters, which comprise the most successful artists in both eras (i.e., higher sales and streams), including Skank and Anitta. In particular, the second one represents the paramount artists, who are all Brazilian, indicating a strong preference for local artists and genres. As a result, the SHA cluster indicates regular success, accounting for over 90% of the artists.

As for the differences, Figure 11 shows a rotation in the preference of the musical genre. Specifically, there was an increase in *Gospel* sales in the Physical Era but not in the Digital one. This fact may be because *Gospel* listeners still consumed physical media by 2015, while audiences from other genres had already migrated to streaming in previous years (until 2016, the PMB methodology still favored physical sales). However, the transition of preference for musical genres over the years is notorious. In the Physical Era, the predominant rhythms were *Axé* (Ivete Sangalo), *Sertanejo* (Zezé di Camargo & Luciano) and *Rock* (Skank); whereas in the Digital Era, the most successful artists (THA) come from one style: *Sertanejo*, with for more than 50% of streamings in late 2020.

Overall, the Digital Era allows the appearance of new popular genres, as well as the decline of previously popular ones. For example, the prevalence of *Sertanejo* is remarkable over time, while *Pop* decreases from 2016 to 2020. Moreover, we highlight the rise of *Forró* in mid-2020 as a popular genre, following the growth of popular artists who have bursted the regional bubble, such as Barões da Pisadinha, Solange Almeida, and Wesley Safadão. Such a significant boost for regional artists may have been enhanced by the remarkable lives of *Forró* and *Sertanejo* artists during the COVID-19 pandemic, showing the music industry's ability to adapt to new realities. In fact, Marília Mendonça had the most-streamed YouTube live worldwide in 2020, with over 3.31 million viewers.

Threats to Validity. Here, a limiting factor is that piracy had a high impact in the music consumption in Brazil, mainly in the late 2000s and early 2010s. Therefore, data collected from PMB may not entirely reflect Brazilian preference in music.⁹ In addition, although we found similar patterns between the Physical and Digital eras, each data source used considers its own success measure, which can cause biased results. Finally, we only consider artists who are recognized as successful (either through their sales or their position in stream rankings). Future work should overcome all such limitations to enhance the results and further advance in the state-of-the-art.

8. Conclusion

Here, we performed a cross-era comparative analysis between physical and digital musical media in Brazil, the largest market for recorded music in Latin America. First, we built artists' success time series to identify hot streak periods for both eras, defined as continuous high-impact bursts. Next, we characterized such periods to understand the dynamics of success among artists from different musical genres. Our results showed that, although there are similarities among all music styles, some genres have meaningful specific patterns for both eras. Therefore, as in other studies in the MIR field, considering music genre information can be relevant for both the predictive and descriptive models. Finally, a profiling analysis uncovered three different clusters in both eras: Spike Hit Artists (SHA), Big

⁹For reference, 52% of the music consumption in Brazil in 2005 came from piracy: <https://bit.ly/PiracyReportIFPI>

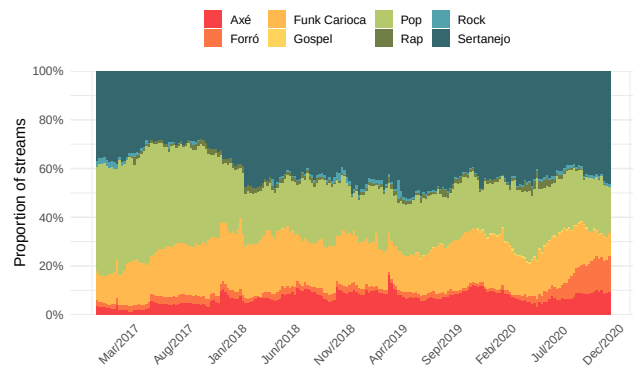
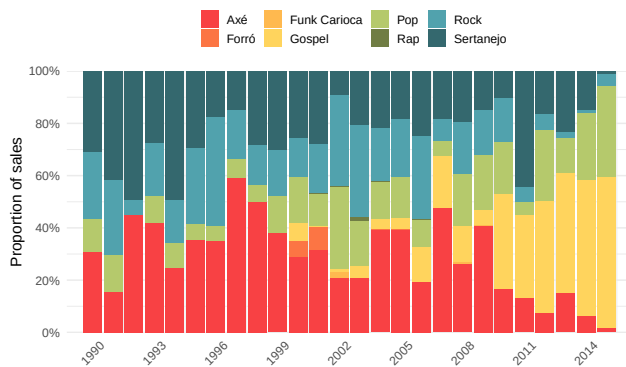


Figure 11: Genre evolution in (left) Physical and (right) Digital Eras.

Hit Artists (BHA), and Top Hit Artists (THA), which acted as class descriptors of successful artists.

Overall, our findings offer meaningful insights for MIR tasks, such as prediction and recommendation. For example, the identified clusters information may be used as features in musical success prediction models. In addition, they may also help in recommending potentially successful musical partnerships and collaborations. Besides helping the scientific community, this work also benefits the music industry as a whole. Analyzing the evolution of artist careers reveals success trends in Brazil from what people consume. Indeed, our results uncover that although Brazilians are connected with international hit songs, they still have a strong preference for local artists regardless of the era. Hence, considering individually regional markets is crucial for better comprehension of specific factors driving musical success. Finally, understanding hot streak periods and success patterns can enhance not only the human element in the music industry (e.g., A&R executives and record labels CEOs) but also people's relation with music. Our findings may help describe the listeners' behavior and musical trends, allowing the music industry to connect people to songs relevant to them.

Acknowledgments. Work supported by CAPES and CNPq, Brazil.

References

- [1] Débora C. Corrêa and Francisco Ap. Rodrigues. A survey on symbolic data-based music genre classification. *Expert Systems with Applications*, 60:190–210, 2016.
- [2] Vítor Shinohara, Juliano Foleiss, and Tiago Tavares. Comparing meta-classifiers for automatic music genre classification. In *SBCM*, pages 131–135, 2019.
- [3] Carlos Soares Araujo, Marco Cristo, and Rafael Giusti. Predicting music popularity on streaming platforms. In *SBCM*, pages 141–148, 2019.
- [4] D. Martín-Gutiérrez et al. A multimodal end-to-end deep learning architecture for music popularity prediction. *IEEE Access*, 8:39361–39374, 2020.
- [5] R Sinatra, D Wang, P Deville, C Song, and A-L Barabási. Quantifying the evolution of individual scientific impact. *Science*, 354(6312), 2016.
- [6] Kiran Garimella and Robert West. Hot streaks on social media. In *ICWSM*, pages 170–180, 2019.
- [7] Lu Liu et al. Hot streaks in artistic, cultural, and scientific careers. *Nature*, 559(7714):396–399, 2018.
- [8] Milan Janosov, Federico Battiston, and Roberta Sinatra. Success and luck in creative careers. *EPJ Data Sci.*, 9(1):9, 2020.
- [9] Leonardo De Marchi and João Martins Ladeira. Digitization of music and audio-visual industries in brazil: new actors and the challenges to cultural diversity. *Cahiers d'Outre-Mer*, 71(277):67–86, January 2018.
- [10] Marcelo Kischinhevsky, Eduardo Vicente, and Leonardo De Marchi. Em busca da música infinita: os serviços de streaming e os conflitos de interesse no mercado de conteúdos digitais. *Fronteiras-estudos midiáticos*, 17(3):302–311, 2015.
- [11] Joel Waldfogel. How digitization has created a golden age of music, movies, books, and television. *Journal of economic perspectives*, 31(3):195–214, 2017.
- [12] Mariana O. Silva, Laís M. Rocha, and Mirella M. Moro. Collaboration Profiles and Their Impact on Musical Success. In *ACM/SIGAPP SAC*, pages 2070–2077, 2019.
- [13] Gabriel P. Oliveira et al. Detecting collaboration profiles in success-based music genre networks. In *ISMIR*, pages 726–732, 2020.
- [14] R Borges and Marcelo Queiroz. A probabilistic model for recommending music based on acoustic features and social data. In *SBCM*, pages 7–12, 2017.
- [15] R de Araújo Lima et al. Brazilian lyrics-based music genre classification using a BLSTM network. In *ICAIS*, 2020.
- [16] Darryll Hendricks, Jayendu Patel, and Richard Zeckhauser. Hot hands in mutual funds: Short-run persistence of relative performance, 1974–1988. *The Journal of finance*, 48(1):93–130, 1993.
- [17] Matthew Rabin and Dimitri Vayanos. The gambler's and hot-hand fallacies: Theory and applications. *The Review of Economic Studies*, 77(2):730–778, 2010.
- [18] Markus Raab, Bartosz Gula, and Gerd Gigerenzer. The hot hand exists in volleyball and is used for allocation decisions. *Journal of Experimental Psychology: Applied*, 18(1):81, 2012.
- [19] Gabriel B. Vaz de Melo, Ana F. Machado, and Lucas R. de Carvalho. Music consumption in Brazil: an analysis of streaming reproductions. *PragMATIZES - Revista Latino-Americana de Estudos em Cultura*, 10(19):141, 2020.
- [20] Eamonn J. Keogh and Michael J. Pazzani. Scaling up dynamic time warping for datamining applications. In *SIGKDD*, pages 285–289. ACM, 2000.
- [21] R Tavenard et al. Tslearn a machine learning toolkit for time series data. *J.Mach.Learn.Res.*, 21:118:1–118:6, 2020.
- [22] Purnima Bholowalia and Arvind Kumar. Ebc-means: A clustering technique based on elbow method and k-means in wsn. *Int'l J. of Computer Applications*, 105(9), 2014.

The 4E's Model of Enactivism through Improvisation within DMIs

Isabela Corintha*, Giordano Cabral

Universidade de Aveiro – 3810-193, Aveiro, Portugal

Universidade Federal de Pernambuco – 50670-901, Recife, Pernambuco, Brasil. ²

isabelaalmeida29@ua.pt, grec@cin.ufpe.br

Abstract

This paper describes how a theoretical framework focused on the 4E's model which describe the mind as fundamentally: *embodied, embedded, extended* and *enactive*, within the paradigm of enactive music cognition can contribute to the design of Digital Music Instruments (DMIs).

From an epistemological perspective, we discuss improvisation within the Western academic music culture through two examples of DMIs created to improvise in a MIDI keyboard. We argue the 4 E's model orientation revealing the fundamentally nature of the embodied musical mind. Ethical and practical possibilities for an enactive music cognition related to improvisation in the context of the skills and needs of 21st are suggested with the goal of helping DMIs designers and musicians to develop approaches based in possibility, imagination, and relationality, rather than in conformity to standardized practices and conventional music pedagogical purposes. Finally, we present two concrete cases of DMIs, and describe how the experience of musical improvisation with them may be seen through the prism of such theories.

1. Introduction

The acquirement of knowledge playing an instrument is essentially internalizing how an action is translated to sound [1]. Different from acoustic instruments, a specifically designed digital musical instrument (DMI) can provide an immersive and embodied musical experience without prior training, unlike the long time it takes a music student to play the basics on an acoustic instrument [2]. Khoury states that if music education is to respond to the skills and needs of 21st-century music learners, innovative learning paradigms must be explored [3].

This paper presents an enactive approach to music education within the 4E's model which describe the mind as fundamentally: *embodied, embedded, extended* and

enactive. In what follows, we demonstrate two DMIs examples and their applicability in the educational practice. The DMIs were both built on a handcrafted plywood structure, which includes all hardware components of an illuminating bar to be placed on a MIDI keyboard controller. It aims to provide visual feedback to novice pianists.

The field of embodied cognition (Varela et al., 1993) within the 4E's model in its potentiality to express the relevance of improvisation in musicianship is outlined with the aim of responding to a broader demand for musical learning of the 21st century [3, 4].

We consider that through innovative learning technologies focused on improvisation, novice pianists and music students may explore their musical skills from the beginning of their training in the instrument [5, 6, 7]

It is important to note that this is a subjective work and present observations and understandings concerning improvisation practice in the theory of embodied music cognition considering enhancing music education.

We conclude by presenting a reflection upon music pedagogy through the 4E's model in improvisational practice within the two presented examples of DMIs.

2. Improvisation within Enactivism

Christopher Small (1998) conceived the term “musicking” to emphasize that listening is as a seminal aspect to express the activity of music. [8, 9]. In this work we associate the enactive approach and the term “musicking” within improvisation. The specific argument that improvisation is essential within music teaching and learning is held by a growing number of music pedagogues [8, 10, 11, 12] and deeply integrates the fundamentals of music into a more comprehensive musicianship [3, 9]. Furthermore, due to the intrinsic nature of improvisation, it is considered as a highly inclusive and cross-cultural practice in which people participate in a more embodied form of music-making than that entailed in repertoire performance. Improvisation is an activity that generally

* Supported by CAPES

involves spontaneity for listening and acting and, in this task is addressed from the perspective of pedagogy.

As stated by Varela et al., (1993) “*The term enactivism was chosen to emphasize that cognition is not the representation of a predetermined world by a predetermined mind, but rather the representation of a world and a mind based on a history of the variety of actions that being in the world performs*”. The enactivist theory does not yet offer a fixed method of assessment, however its ideology proposes a set of concepts that illustrate the creative practice of self-reflection and self-assessment. Taking this into consideration, the enactivist approach reveals living cognition as fundamentally improvised [4, 13, 14, 15].

2.1 4E's of Cognition & Music Pedagogy

Van der Schyff states that enactivism can be defined, broadly speaking, in two ways. First it may be approached according to cognition. Recently, these have been referred to as the “4E’s,” which describe the mind as fundamentally: *embodied, embedded, extended and enactive*, and have characteristics in common that led them to be grouped in this way.

The 4E’s model seek to open perspectives on the conception of the human mind by exploring elements other than the brain. The 4 E’s include the body, the environment and even technologies. According to this view, cognition depends on the body - in addition to the brain - to shape and limit cognitive processes: the mind is *embodied*. While it emerges from the body as a whole, in addition to the brain, cognition is also situated, since it also needs the environment to emerge, it is *embedded* [16]. Once knowledge is *embedded*, it depends a lot on the physical and socio-cultural milieu; cognition is *extended* to the environment. The environment defines the cognitive load among other beings and technologies [9, 17]. Finally, based on these three previous principles, there is an *enactive* knowledge, since knowledge is formed through co-adaptive couplings between beings and their environment. Although these principles overlap and build, they are often referred to collectively as 4E’s model of cognition [15, 16, 17, 18].

As improvisation is argued as a situated practice that embraces adaptivity, contingency, and the unexpected [7], an exploration of improvisation through the 4E’s model may reveal new perspectives on teaching, learning, and assessment that could have profound implications for the future of musical education [12, 18].

3. DMIs – Music apprenticeship

Exploring and manipulating an instrument with somewhat arbitrary actions can lead to unexpected results and all these movements are comprehended in the learning music process [1].

McPherson et al., write that a designed digital musical instrument (DMI) can provide an immersive and embodied musical experience without prior training, opposed to the hundreds of hours needed to achieve basic tone production on many acoustic instruments [2].

The design of musical instruments to make performance accessible to novice musicians is a goal that predates digital technology [2]. Although the discussion of the DMIs design within skills development, practice, mapping and the different layers of feedback is mostly focusing on the art of performance, a large number of DMIs are designed with varying degrees of applicability in the educational practice [19].

The diversity in the framework of DMIs design can adopt several typologies related to a range of categories. For example: Inter-actors involved in a performative ecology using a DMI; the interaction input control (e.g., gestures, gloves, keyboards, mobile phones et al.); the control parameters (e.g., pitch, duration, dynamics, timbre, vibrato, other audio effects) and the typology of the system, ranging from sequenced to generative responses [19] are dimensions adopted from [20, 21, 22].

A digital musical instrument is constituted by more dimensions than just the physical and that these dimensions are also capable of eliciting perceptual experiences and even insights [23]. It embeds musical culture and musical work practices considering that its framework is designed within a wide range of dimensions and the result will be informed by the conceptual capabilities and contextual choices of its creator [23].

Musicians and designers have several motivations to build their instruments, among them: bring greater embodiment to the activity of performing and producing electronic music; improve audience experiences of DMI performances; sound synthesis development; build responsive systems for improvisation [24] and promoting new pedagogical approaches [19].

The focus on improvisational musicianship using DMIs for education, responds to a broader demand to 21st century apprenticeship [2, 3, 25]. If the body plays a key role in determining musical learning [26], so does the socio-material and cultural environment in which it is *embedded*.

DMIs expand the traditional acoustic instruments and challenge the musical practice towards new corporeality, materiality, control and feedback [19]. An enactive approach applied to DMI design comprises users and interfaces immersed in a shared autonomy system, so both co-evolve from the experience of interaction [27]. The machine ability regulates the control input which include, for example, gestures, tangible user interface, sensors, keyboard, sound, joysticks, gloves, VR glasses, semi-haptic and haptic interfaces, and respond to this through its actuator [19]. An autonomous DMI is represented by the interface and the performer coupled through their sensors/actuators resulting in an embodied system [27].

Dobrian & Koppelman write that in trying to design an instrument that will enable expression, it is necessary to consider how the performer will provide musical expression, notably how the performer’s gesture will affect the sound [28]. Figure 2. represents the flow of information between the source and the sensorimotor gestural feedback within an enactive interface.

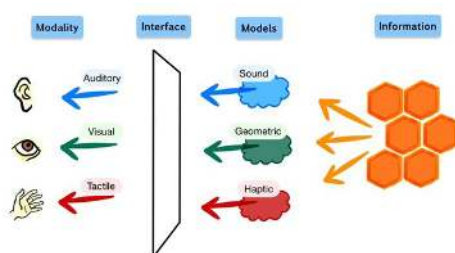


Figure 1. An enactive interface based on a draft by Monica Bordegoni, 2010.

The 4E's model in DMIs is not only related to the feedback between action and perception, it is also linked to how this sensorimotor loop, is translated to the symbolic domain under which it operates the system control and instrument processing [27]. The communication between the performer's gesture and the DMI sound response will determine the cohesion of the established temporary unity. Technologically enhanced listening accompanied by gestural feedback may become a tool of great significance for learners in the 21st century [3, 9, 26]. Existing between the acts of musical study, the practice of performing and listening through DMIs [29, 30] may provide music learners with a profound perception and sensibility for music that can shift the way they get into listening, improvising, performing and any activity that involves an aural sensibility [2, 3].

4. The 4E's model within music improvisation

Thor Magnusson writes that "the analysis of digital music systems has traditionally been characterized by a phenomenological approach. The focus has been on the body and its relationship to the machine, often neglecting the system's conceptual design" [30]. His work investigates the epistemic nature of digital musical instruments dimensions. From an epistemological or music-theoretical perspective, his work addresses the culture-theoretical aspects that so prominently define their nature within eight axes: *Expressive Constraints*, *Autonomy*, *Music Theory*, *Explorability*, *Required Foreknowledge*, *Improvisation*, *Generality* and *Creative-Simulation*. The *Improvisation* axis indicates the degree to which the instrument lends itself to free improvisation, how responsive it is, how open for changes in real time performance and how quickly can it be adapted to those.

Considering the improvisation axis from the epistemic dimension space by Thor Magnusson, which is based on the work from Birnbaum et.al [31] and the DMI control input addressed through bodily motion, we propose an overview of digital musical instruments. In order to provide novel insights that may help inspire a richer understanding of what musical learning through improvisation within DMIs entails, the following aspects (or questions) concerning the 4E's model are described:

- *Embodied*: focuses on the body-instrument relationships and understandings outlined by the DMI's design. An *embodied* account describes music perception and musical action not as divorced, an intuition for

melodic/harmonic/rhythmic involving more than the brain [9].

- *Embedded*: represents the amount of music possibilities explored and developed in physical, sonic, historical, social, cultural and gendered world(s) [15]. How can a DMI be an effective improvisational tool considering the environment and all musical genres such as the carnatic music, cumbia, or simply bossa nova?

- *Enactive*: represents how much of depth the DMI holds within the capabilities-in-action. This factor regards how the engagement with the instrument affects the learning curve [17]. How can the DMI transform the ways we engage with the world musically, sonically, socially, emotionally and so on?

- *Extended*: Specifies how our creative possibilities can be enhanced through interactions with co-performers, technologies, and other non-organic ecological factors. How can a DMI help to facilitate the musical creative development? An *Extended* phenomenon emerges in relation with devices and environments that co-constitute music-like behaviors (and not only "afford" them) [9, 32].

The 4E's model applied to DMIs design comprises users and interfaces immersed in a shared autonomy "autopoietic" system, so both co-evolve from the experience of interaction. An autonomous DMI is, therefore, an embodied system that satisfies its internal goals through its actions in the environment [27]. The enactive approach provides new possibilities for DMIs design considering the human interaction in the social cultural milieu.

5. AMIGO and AM-I-BLUES within the context of musical improvisation and the 4e's model

In this section, we illustrate how two examples of digital musical instruments can be seen through the lens of the 4E's model. Naturally, this is a subjective approach, and would ideally be performed by way of user surveys [31]. We took one of the two examples from a catalogue compiled across all editions of the International Conference on New Interfaces for Music Expression (NIME) with varying degrees of DMI applicability in the educational practice [19]. The second example integrates the existing technology from the first one: A handcrafted plywood structure which includes all hardware components of an illuminating bar to be placed on a MIDI keyboard.

AMIGO is a digital musical instrument that promotes embodied and enactive sense-making in music theory environment. The DMI main aim is to learn to improvise melodies [30]. Gradation colors (from yellow to red) guides the user on a physical keyboard mounted with a LED stripe. The intelligence of the system relies on its capability to derive probabilistic models of note transitions from existing musical examples encoded in the MIDI format. A computer screen translates user actions into musical notation, which can be edited and retrieved at a later stage.

From the 4E's model perspective, *AMIGO* can be considered bordering on an *embodied* instrument since it

promotes continuous integration of sensorimotor activity (action-as-perception). We consider that it cannot be thought of as an *embedded* tool since it is related only to Western music formalisms, hence it does not boost adaptive behavior within the socio-material and culture niche we may inhabit. Although *AMIGO* can be situated within a world with which it interrelates, this work is addressed to a worldwide musicianship. It is a DMI with *enactive* properties since it offers capabilities-in-action for the learning music process through improvisation and music theory contents as shown in Figure 5. Lastly, it has extended aspects since it is an intuitive tool for the creation of musical structures. Its main aim is to stimulate the learning musical process through improvisation; in our rough analysis we consider it as an extremely *extended* DMI.



Fig. 2. AMIGO's interface displaying the music notation feedback and its MIDI controller mounted with a LED stripe.

Finding the right chords to adjust a melody or the right melody to adjust a chord progression is often in the improviser's task. *AM-I-BLUES* [25] is a DMI that integrates two existing technologies: the illuminating controller interface of *AMIGO* [6] with the generative jazz model developed for the *MyJazzBand* installation [34]. The resulting DMI aims to introduce novice pianists in the improvisation of blues and jazz melodies by guiding them via the visual feedback of an illuminating keyboard. *AM-I-BLUES* fosters a musical understanding and expressiveness of the mechanics and theory of melodic improvisation within the blues/jazz idiom without music theory knowledge. Yet, the DMI was designed to consider some prior performative skills on the keyboard, as the target user is mostly restricted to novice pianists.



Fig. 3. The Illuminating Keyboard. The plywood structure holding the LED array docked to the MIDI Keyboard controller.

The major novelty of *AM-I-BLUES*, in comparison to existing illuminating keyboards, is the use of a generative system, which provides harmonic sequences on-the-fly, to which multiple optimal solutions for note selection are provided for melodic creation.

The control input of this system, the MIDI keyboard, are in our context, a *embodied* framework of a DMI, promoting intense interactive feedback between action and perception. This DMI is focused on the blues/jazz idiom and it can be situated within a world with which it interrelates, but once it is focused on Western music context, in our subjective analysis it is not *embedded*. The generative jazz model provides a way of interacting with the music including implied structure of beats and tempo for learning to improve guided by the system; therefore, we hold an opinion that it is an *enactive* system. As the system stimulates adequate motion and creative potential, we may consider it an *extended* DMI. The system is presented on Figure 3.

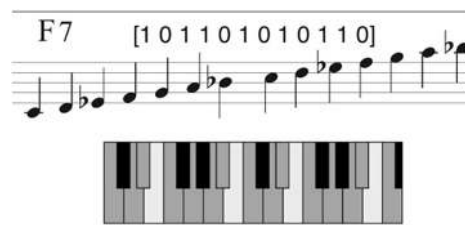


Fig. 1. One example of generated scales corresponding to the first two chords of the blues progression (F7).

The active notes, represented by a "1" in the vector, correspond to the correct notes suggested by the generative system for the Lead instrument and are illuminated in the LED bar in red, while the inactive notes are lit white (see Fig. 3).

6. Discussion

Toward the aim of a meaningful music pedagogy through the 4E's model within DMIs, we suggest designers and musicians a reflection upon the following questions:

- *Embodied* - How can the sonic/musical result of the DMI developed from the body/mind movement provide new perceptions and experiences that provoke melodic, harmonic and rhythmic intuition [12]?

- *Embedded* - Would it be possible to improvise in another milieu with other instruments of a given culture with the DMI? What roles does the DMI in different socio-cultural environments? How a DMI within the environment could be efficient to facilitate other ecological factors of musical categories, including rhythm, pitch, harmony, voice, variation and forms?

- *Enactive* - Which contributions have the DMI to improve the capabilities-in-action of the sensorimotor capacity to improvise?

- *Extended* - What are the DMI creative possibilities to enhance or make possible interactions with co-performers, technologies, and other non-organic ecological factors [13]? How the DMI can enhance the creative development?

7. Conclusions

We believe that the underlying theoretical and conceptual foundation of the 4E's model within the design of DMIs can contribute to the contemporary social challenges supporting the skills and needs of 21st century music learners. The enactive music cognition to improvisation detailed above does not offer a fixed method of assessment; nevertheless, we hope that it will resonate with a rich pool of ideas and research for DMIs designers and musicians.

8. References

- [1] R. H. Jack, J. Harrison, F. Morreale, A. McPherson. Democratising DMIs: The Relationship of Expertise and Control Intimacy. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2018. NIME '18, June 3-6, 2018, Blacksburg, Virginia, USA.
- [2] A. McPherson, F. Morreale & J. Harrison, 2019. Musical Instruments for Novices: Comparing NIME, HCI and Crowdfunding Approaches. Chapter 12. S. Holland et al. (eds.), *New Directions in Music and Human-Computer Interaction*, Springer Series on Cultural Computing, DOI: 10.1007/978-3-319-92069-6_12
- [3] S. L. Khoury. *Improvise: Research-Creation of a Framework and Software Prototype for Creative Music Learning with Technology*. Ph.D. Thesis, University of Montreal, Quebec, Canada, 2017.
- [4] F. J. Varela, E. Thompson, and E. Rosch. *The embodied mind: cognitive science and human experience*. MIT Press, Cambridge, MA, 1991.
- [5] L. Hayes, 2019. Beyond Skill Acquisition: Improvisation, Interdisciplinarity, and Enactive Music Cognition. DOI: 10.1080/07494467.2019.1684059
- M. Bordegoni, 2010. Exploitation of Designers and Customers' Skills and Creativity in Product Design and Engineering. Pages 63-85. In Shuichi Fukuda (ed.). *Emotional Engineering: Service Development*. Springer. p. 76.
- [6] A. Marquez-Borbon. Collaborative Learning with Interactive Music Systems. In *Proceedings of the International Conferences on New Interfaces for Musical Expression NIME'20*, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham, United Kingdom, 2020.
- [7] D. Van der Schyff. *Improvisation, Enaction, and Self-Assessment*. Oxford: The Oxford Handbook of Philosophical and Qualitative Assessment in Music Education. Edited by David J. Elliott, Marissa Silverman, and Gary E. McPherson, 2019.
- [8] C. Small, (1998). *Musicking: The meanings of performing and listening*. Middletown, CT: Wesleyan University Press.
- [9] J. Gutierrez. An Enactive Approach to Learning Music Theory? Obstacles and Openings. Edited by: Dylan van der Schyff, University of Melbourne, Australia. *Frontiers in Education*, 2019.
- [10] P. S. Campbell. Learning to Improvise Music, Improvising to Learn Music. In *Musical improvisation: Art, education, and society Urbana - IL - 119-142*, University of Illinois Press, 2009.
- [11] É. Jaques-Dalcroze. *Rhythm, music and education*. New York, NY: G. P. Putnam's Sons, 1921.
- [12] R. K. Sawyer (2000). Improvisation and the creative process: Dewey, Collingwood, and the aesthetics of spontaneity. *The Journal of Aesthetics and Art Criticism*, 58(2), 149-161.
- [13] J. M. Loaiza. Musicking, embodiment and participatory enaction of music: outline and key points. *Connection Science*, 2016. Volume 28, Issue 4, p.410-422. DOI: 10.1080/09540091.2016.1236366
- [14] J. Mingers. The cognitive theories of Maturana and Varela. *Systems Practice* 4, 319-338, 1991. DOI:10.1007/BF01062008
- [15] D. Van der Schyff. *Improvisation, Enaction, and Self-Assessment*. Oxford: The Oxford Handbook of Philosophical and Qualitative Assessment in Music Education. Edited by David J. Elliott, Marissa Silverman, and Gary E. McPherson, 2019.

- [16] C. Leporace, Espelhos do mundo? Uma perspectiva da percepção humana a partir de ideias da mente enativa e da mente estendida. *A Mente Humana para Além do Cérebro*. Copyright: 2019, Instituto de Psicologia Cognitiva, Desenvolvimento Humano e Social da Universidade de Coimbra, IPCDHSUC.
- [17] A. Schiavio, and D. van der Schyff. 2018. “4E Music Pedagogy and the Principles of Self- Organization.” *Behavioral Sciences* 8 (8): 72. Open Access Journal. Published: 9 August 2018. DOI:10.3390/bs8080072
- [18] D. Van der Schyff, A. Schiavio & D. J. Elliott. Critical Ontology for an Enactive Music Pedagogy. *Action, Criticism & Theory for Music Education* ISSN 1545-4517. Volume 15 Number 5, October 2016. Vincent C. Bates, Editor Brent C. Talbot, Associate Editor
- [19] M. Pessoa, C. Parauta, P. Luís, I. Corintha & G. Bernardes. Examining Temporal Trends and Design Goals of Digital Music Instruments for Education in NIME: A Proposed Taxonomy. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2020. Birmingham, UK. *NIME'20*, July 21-25, 2020.
- [20] B. Bongers. Exploring Novel Ways of Interaction in Musical Performance. *ACM Trans.* In CC '99: Proceedings of the 3rd conference on Creativity cognition, pages 76–81, 1999.
- [21] J. Drummond. Understanding Interactive Systems. *Organised Sound*, 14(2):124–133, 2009, R. Rowe. Interactive music systems: Machine listening and composition. MIT Press, 1993.
- [22] T. Magnusson. Musical organics: a heterarchical approach to digital organology. *Journal of New Music Research*, 46(3):286–303, 2017.
- [23] J. C. Schacher. Hybrid Musicianship - Teaching Gestural Interaction with Traditional and Digital Instruments. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2013. *NIME'13*, May 27 – 30, 2013, KAIST, Daejeon, Korea.
- [24] G. Emerson & H. W. Egermann. Exploring the Motivations for Building New Digital Musical Instruments. *Musicae scientiae*, 2018. ISSN 1029-8649. DOI: 10.1177/1029864918802983
- [25] I. Corintha, L. Outeiro, R. Dias, G. Bernardes. AM-I-BLUES: An Interactive Digital Music Instrument for Guiding Novice Pianist in the Improvisation of Jazz Melodies. *Advances in Design, Music and Arts*, Vol. 9, 689-698. Springer, 2020.
- [26] M. Rodger, P. Stapleton, M. van Walstijn, M. Ortiz & Laurel Pardue. What Makes a Good Musical Instrument? A Matter of Processes, Ecologies and Specificities. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2020.
- [27] G. Rimoldi & J. Manzolli. Enactive Framework for Design of Digital Music Interfaces. In: *Proceedings of the 2nd International Conference on New Music Concepts - ICNMC*, 2016. Corpus ID: 207815887
- [28] C. Dobrian and D. Koppelman. The ‘E’ in NIME: Musical Expression with New Computer Interfaces. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2006.
- [29] A. Marquez-Borbon & J. P. Martinez Avila. The problem of DMI adoption and longevity: Envisioning a NIME performance pedagogy. In *Proceedings of the International Conferences on New Interfaces for Musical Expression (NIME)*, 2018.
- [30] T. Magnusson. An Epistemic Dimension Space for Musical Devices. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2010. *NIME2010*, June 15-18, 2010, Sydney, Australia.
- [31] Birnbaum, D., Fiebrink, R., Malloch, J., and Wanderley, M. Towards a Dimension Space for Musical Artifacts. In *NIME 2005 Proceedings*. 2005.
- [32] D. Van der Schyff, A. Schiavio, A. Walton, V. Velardo & A. Chemero (2018). Musical creativity and the embodied mind: Exploring the possibilities of 4E cognition and dynamical systems theory. *Music & Science*. Volume 1: 1–18 DOI: 10.1177/2059204318792319
- [33] I. Corintha, G. Cabral, G. Bernardes. AMIGO: An Assistive Musical Instrument to Engage, Learn and Create Music. In: *Proceedings of the international conference on New Interfaces for Musical Expression*, 2019. *NIME'19*.
- [34] Dias. R.: Interfacing Jazz: A Study in Computer-Mediated Jazz Music Creation and Performance. Ph.D. Thesis, University of Porto, Portugal, 2018.

An open source platform to assist the creation of group playlists through artificial intelligence algorithms

Flaviano Dias Fontes^{1*}, Giordano Ribeiro Eulalio Cabral¹, Geber Lisboa Ramalho¹

¹Music – Centro de Informática / Universidade Federal de Pernambuco (UFPE)
Av. Jornalista Anibal Fernandes, s/n – 50.740-560 Recife, PE

Abstract. Recommendation systems are a constantly expanding study area, with applications in various fields such as e-commerce, films, music to promote the user's suggestions. When we talk about music, we have more than 20 years of studies trying to solve the problem of a good generation of playlists that maximizes the satisfaction of a larger number of listeners. For automated automatic playlist generation methods focusing on a user group, we have the collaborative filter as a more assertive method to get the user's not likely, to improve the performance of group recommendation algorithms we store the preferences of users. Especially I did not like it by placing the availability of using this data as an algorithm input parameter. The platform described in This paper is intended to facilitate testing between these recommendation systems, standardizing data entry, and facilitating requests. The use of GraphQL as a framework associated with Apollo as a library, greatly facilitates the integration of these APIs, as the separation of data sources makes it possible to associate Spotify data with Deezer or Apple Music data, these data are stored in the database of the connection, so that in future requests it will no longer be necessary to consult the Spotify API, thus facilitating the consumption of data from the artificial intelligence algorithms, as well as a possible sharing of songs between services, since all services have an ISRC code to identify the songs.

1. Introduction

The music market has suffered several changes over the years and after a recent and long decline in revenues, the industry has finally managed to rejoice in revenue through streaming services, becoming the largest source of revenue, taking Spotify as principal service [1].

This way of consuming is based on the availability of songs in the order of millions of copies, for streaming services. Some user-related variables are taken into account in the recommendation systems to define their musical taste, as their level of experience, their location, and their personality traits, however for user consumption has only released the songs he likes, a set of songs that the platform understands that he likes or just the playlists he created, but is not available what the user does not he likes [2]. It is easily found in the Application Programming Interfaces (API) of the Streaming Services, [3] the methods for querying playlists but with the limitation of not having the consultation of the songs that the user does not like.

*Supported by CNPq.

2. Music Recommender Systems Evolution

Recommendation systems are a set of software and techniques that suggest to the user items to be used for it. Suggestions refer to decision-making processes, such as what news reads, which product buy or music listen [4].

These systems solve the problem of information overload and help users choose from the various options in their daily lives. Recommendations systems capture previous users' preferences and generate an available list of items to meet the user [5].

Recommendation of groups for groups is a subject that has been studied, an example is the Smart Radio in 1996 that was a web application in the client-server format on which it allowed users to automatically filter and collect the songs, with User dislike capture, another important point to stand out is the automatic extraction of features making it possible to improve content-based recommendation. [6]

In 1998, MusicFX enhanced the capture of information to recommend musical genres through collaborative filters, on a scale described by I love (+2), I like (+1), I do not care (0), I do not like (-1) and I detest (-2), the algorithm adjusts the recommendations based on the evaluation of the genres of users present in the environment, described in the article as a gymnastics. [7] This approach to capturing directly does not like the user is a facilitator but through the main consultations, the APIs are not available to third parties what creates our need to capture this information about the user's dislike [8] [9] [10].

Flytrap in 2002 was a recommendation system that reproduces songs that users like, through music that a user plays on their computer it loads the data to the FLY-TRAP database. Unlike MusicFX, the information sent are recovered through the playback of the track and these meta-data are, for example, the genre and the artist stored this information linked to the user who played. [11]

The JMusicGrouPrecommender merges recommendations from individual users, adds personal evaluations of users, and finally generates group preferences to generate recommendations. In this work, a fusion technique, a technique for generating group preferences, four aggregation techniques, and a hybrid technique including mixing and aggregation are proposed. Each technique is evaluated based on the satisfaction of each user who is within the group. Each song contains some attributes such as artist, album, language, and gender [12].

This article it was describing are proposals, a fusion technique, a technique for generating group preferences, four aggregation techniques, and a hybrid technique including mixing and aggregation. Each technique is evaluated based on the personal satisfaction of all team members. Each song contains some attributes, such as artist, album, language, and gender. These attributes are used to infer indefinite membership preferences. Among all the algorithms proposed in this article, it is reported that the hybrid technique has high rates of individual satisfaction among the users of a group. Compared to Automatic Consensus Selection, this template offers users the option to have at first a set of items [13] [5].

The problem when data is spaced is addressed in group recommendation systems in different ways. It is important to note that traditional group recommendation techniques do not produce better recommendations when the user-item relationship is sparse, and your preferences are unknown. [5]

The possibility of calculating the similarity of the item-item and computed similarities are used to improve existing memory-based recommendations techniques. The degree of similarity is a decimal attribute that is between 0.0 and 1.0, which is added in this template to all pairs of item-item. Initially, the support technique of the Support Vector Machine (SVM) employed in this model is compared to other regression models, and has been discovered that SVM surpasses other models. The proposed model surpasses the latent factor model, which is general considered the best method to use among memory-based models. [14]

3. Streaming Available Information

Using as a matrix of characteristics of the Flytrap and JMusicRecommender features we have as a reference for information required to create music recommendation systems, the artist, the album, and the language. If the approach is a collaborative filter the preferences of users of a particular item must be stored in a database. [11] [12] For collection of this information the REST access points of Spotify, Apple Music, and Deezer as an Apollo data source, a GraphQL implementation that provides a data chart layer that connects applications to the cloud. [15] [8] [3]

3.1. Spotify

Queries in this API occurs through REST requisitions by following the URI standard. [16] The consumption of the Spotify API has the need for a free developer account and you can see user playlists list more played songs among other methods. By consulting certain music and recovering a lot of information such as name, platform popularity, artist, duration in MS, markets that music is available, album cover, and the International Standard Recording Code (ISRC). To get the genre only with the artist, but for being a list for the artist the selection of music can be deficient, as it is possible for an artist to evolve musically during his career and not get stuck in a style necessarily. [3].

3.2. Apple Music

Access to the MusicKit API, you need to have an Apple developer account to be able to make the requisitions, the account is required so that you can generate the access keys to the application. [17] As the API is still in beta, some features are not as ripe as other APIs, but it is possible, for example, Search music by IRC, by name approach, in addition to user playlists but user information If a new requisition of access to data is required. The information that API returns on music are artist name, album name, cover art and colors, composer name, list of genres that music has, duration in milliseconds, launch date, and an ID for easy localization. [8]

3.3. Deezer

The consumption of information through the Deezer API is due to your API, to access this API requires a developer account that is free. The API offers the search method by music name, the return of the track search method contains information from the album, the artist, gain, bpm, preview with 30 seconds, ISRC, duration, title plus other information related to platform as rank. [9]

4. Proposal

The conception of a playlist for just a minimally accurate user of the information than he likes and from this information recommend what other users who also like that item consume. However, when we want to recommend for groups we come across the situation of what a user likes the other hates and riding a playlist that will be consumed by all users, for streaming we still have the availability barrier of particular music on the platform that the user consumes. To minimize this problem is proposed an open-source platform for group playlists generation through community-developed algorithms to supply from the lack of information between platforms, such as audio characteristics that are available only in Spotify [3], Or even the search for similarity available at Deezer [9] and Apple Music [10].

Only the Spotify API returns the user's most listened songs, the Deezer API has a list of recommendations with what the user is interested in, know as Radio, the Apple Music API has several restrictions, but because being in beta is expected This behavior [3].

All 3 APIs [8] [9] [3] return to the user the ISRC information, with the Apple Music API search methods by ISRC or by the name of the band returning what makes it easy to build our database. Common information between all APIs, is the artist name, album name, track name, ISRC, and duration. In order to be able to use any machine learning algorithm, more information is required in the vector of features, this additional information is only available through the Spotify API being them danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence and tempo. [18]

5. Architecture

It is an open-source client-server platform based on GraphQL, a query-based query language developed by

Facebook in 2012 and publicly released in 2016 [19]. An alternative to the REST architecture, but with the ability to query REST APIs. The server is responsible for processing user requests through queries sent in client applications. These requests are intended to collect information from the streamers that the client has.

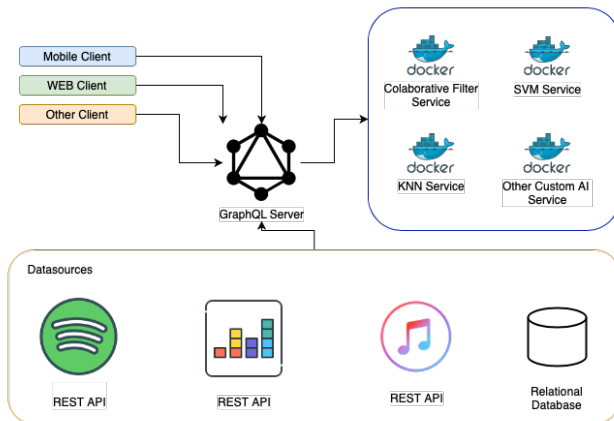


Figure 1: Architecture

6. Modules

6.1. GraphQL Server

For the GraphQL server, we use the Apollo library, officially recommended for applications in Node.JS, this module is responsible for being the clients' access point, the connection with other REST APIs, and the handling of different data sources. Your query is done through queries executed by the client, returning only what is necessary for each request made. [15]

6.2. GraphQL Client

Using Apollo as a GraphQL library makes it easy to integrate with iOS, Android, and Web platforms, as there is a repository for each client, thus facilitating integration, even if it is a React.JS application as a native Android or iOS application, just needing to install the library on the desired platform. Other services can also consume the API through the application's access point using the web client.

6.3. Data Sources

The data sources we use for the platform are Spotify, API where we get the audio features, Deezer and Apple Music, where we get tracks similar to that title, to save the relationships between players in the platform's database, making it easier the execution of the recommendation algorithms, making the default input.

6.4. Services

The services are run inside Docker containers in order to have the necessary isolation, having a playlist as input standard, but also a certain algorithm can take into account the mood of the participants, for example, using GraphQL it is

possible to do this request passing the necessary additional information. For the routing of containers, a web server is used, such as NGINX, also as an image, for example. These services, in addition to processing artificial intelligence algorithms, it is also possible to manipulate data between stream services APIs, such as playlist manipulation.

7. Future perspectives

We will test this platform to validate whether how standardized data is made available allows for some gain in music recommendation systems, we will also test whether the use of the Apollo library brings any improvement in the consumption of third-party API data compared to direct consumption through consumption via REST. Test whether the addition of recommendation models with neural networks or deep learning is easily coupled to the platform.

References

- [1] Arnt Maasø and Anja Nylund Hagen. Metrics and decision-making in music streaming. *Popular Communication*, 18(1):18–31, 2020.
- [2] Martijn Millecamp, Nyi Nyi Htun, Yucheng Jin, and Katrien Verbert. Controlling spotify recommendations: effects of personal characteristics on music recommender user interfaces. In *Proceedings of the 26th Conference on user modeling, adaptation and personalization*, pages 101–109, 2018.
- [3] Spotify web api. <https://developer.spotify.com/documentation/web-api/>. Accessed: 2020-12-22.
- [4] Francesco Ricci, Lior Rokach, and Bracha Shapira. Introduction to recommender systems handbook. In *Recommender systems handbook*, pages 1–35. Springer, 2011.
- [5] Sriharsha Dara, C Ravindranath Chowdary, and Chintoo Kumar. A survey on group recommender systems. *Journal of Intelligent Information Systems*, 54(2):271–295, 2020.
- [6] Conor Hayes, Pádraig Cunningham, Patrick Clerkin, and Marco Grimaldi. Programme-driven music radio. In *Proc. of the ECAI*, volume 2, 1996.
- [7] Joseph F McCarthy and Theodore D Anagnost. Musicfx: an arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the 1998 ACM conference on Computer supported cooperative work*, pages 363–372, 1998.
- [8] Apple Music documentation. <https://developer.apple.com/documentation/applemusicapi/>. Accessed: 2021-05-12.
- [9] Deezer api documentation. <https://developers.deezer.com/>. Accessed: 2021-02-15.
- [10] MusicKit documentation. <https://developer.apple.com/documentation/musickitjs/>. Accessed: 2021-05-12.
- [11] Andrew Crossen, Jay Budzik, and Kristian J Hammond. Flytrap: intelligent group music recommendation. In *Proceedings of the 7th international conference on Intelligent user interfaces*, pages 184–185, 2002.
- [12] Ingrid A Christensen and Silvia Schiaffino. Entertainment recommender systems for group of users. *Expert systems with applications*, 38(11):14127–14135, 2011.

- [13] Mike Gartrell, Xinyu Xing, Qin Lv, Aaron Beach, Richard Han, Shivakant Mishra, and Karim Seada. Enhancing group recommendation by incorporating social relationship interactions. In *Proceedings of the 16th ACM international conference on Supporting group work*, pages 97–106, 2010.
- [14] Sarik Ghazarian and Mohammad Ali Nematbakhsh. Enhancing memory-based collaborative filtering for group recommender systems. *Expert systems with applications*, 42(7):3801–3812, 2015.
- [15] Apollo framework. <https://www.apollographql.com>. Accessed: 2021-06-20.
- [16] Wei Zhou, Li Li, Min Luo, and Wu Chou. Rest api design patterns for sdn northbound api. In *2014 28th international conference on advanced information networking and applications workshops*, pages 358–365. IEEE, 2014.
- [17] Creating an apple music api token for musickit js. <https://leemartin.dev/creating-an-apple-music-api-token-e0e5067e4281>. Accessed: 2021-06-20.
- [18] Audio features object. <https://developer.spotify.com/documentation/web-api/reference/#object-audiofeaturesobject>. Accessed: 2021-02-15.
- [19] Ruben Taelman, Miel Vander Sande, and Ruben Verborgh. GraphQL-ld: linked data querying with graphql. In *ISWC2018, the 17th International Semantic Web Conference*, pages 1–4, 2018.

Art Papers

Playing Time-Variant Audio Feedback Networks

Adriano Claro Monteiro

Escola de Música e Artes Cênicas – Universidade Federal de Goiás
Campus Samambaia, Rodovia GO-080, Km 3/4 – 74001-970, Goiânia, Goiás

adrianomonteiro@ufg.br

Abstract

This article presents practical and artistic contributions to the field of computational music systems based on audio feedback networks. The works presented here have been used as instruments for music creation in the author's artistic practice. The article begins with an introduction to the research field on feedback and self-organized music systems. Later on, two systems are presented: the first is a network of cross-modulated sinusoidal oscillators (by frequency modulation), and the second is a network of transformation processes of pre-recorded sound samples.

1. Feedback systems in music

Since the 1960's, audio feedback has been inventively used in experimental approaches to music creation, from the use of Larsen effect [1] by rock'n'roll guitar players in bands such as The Beatles and The Jimmy Hendrix Experience, to the creation of analog feedback circuitry in the avant-garde/experimental art-scene, in works by Robert Ashley, John Cage, David Tudor, Gordon Muma and Alvin Lucier [2]. Alongside these experimental (maybe more empirical and intuitive) approaches, a theoretically motivated use of feedback in the art of the 1960's took place under the influence of cybernetics [3] and system theory [4] – that were popular at the time – for instance in works by Nicolas Schöffer or Roland Kayn [5] [6]

More recently, new approaches to the creation of autonomous generative music systems based on feedback networks have been relying on computational methods. An important technical addition provided by computers lies in their capacity to analyze incoming data in order to provide informed algorithmic responses to environmental stimuli [7][8][9][10][11][12]. Some of the authors of these recent feedback based works also report influence from cybernetics and system theories and include to their theoretical framework the concept of autopoiesis introduced in the 1980's by thinkers such as Francisco Varela and Humberto Maturana [13], claiming an eco-systemic paradigm to define their art works[7][8] [14][15].

Concepts usually associated to generative autonomous feedback systems – that describe their features, behavior and outcome – are: causal circularity, non-linearity, self-organization, emergency, complexity and synergy [5]. In summary, these systems tend to exhibit *non-linear* behavior, which means that their output is not directly proportional to their input, yielding *complex* results with *emergent* characteristics that, even though are

based on deterministic process, are difficult to predict. *Self-organized* is the term used to describe autonomous systems that exhibit global patterns of organization emerging from the interaction between its parts in parallel and distributed processes, and adapting itself to external conditions [16]. The term *synergy* describes the coupling of parts of a system under interaction of reciprocal influence in which cause and effect are mutually dependent (i.e. causal circularity). That produces an outcome not reducible to the sum of its functional parts (here, once again, the *emergent* feature is described).

Previous publications by Sanfilippo and Andrea [5], and Kollias [17] provide an comprehensive review of musical works in this field, as well as insights to understand their similarities and differences regarding technical and artistic characteristics.

As a contribution to this field of artistic research, this paper presets two original implementations of audio feedback networks that have been used as instruments for music composition and improvisation by the author. The systems differ in their architecture and method for sound generation: the first is based on sound synthesis by Frequency Modulation (FM) [18] in a directed cross-modulation interaction, and the second is based on transformation processes of pre-recorded sound samples and adopts a interaction mediated through audio analysis algorithms.

2. Time-variance and instrumental properties

Sanfilippo [11] points out two other categories to understand feedback systems: *time-invariant* vs *time-variant*. According to the author, a *time-invariant* system “*performs the same operations at all times*”. Its output can be dynamical, as it changes over time, yet its internal state (its operations) is static. A time-variant system allows for changes in its operations over time, which will likely lead to changes in its output.

The term *adaptive*, also discussed by Sanfilippo, is generally applied to describe systems that change their internal state in function of an input received from its external environment in an attempt to achieve some sort of balance with it (usually through combined process of positive and negative feedback), or to better accomplish an explicit or implicit goal [19][20]. In this sense, an *adaptive* system is also a type of *time-variant* system.

The two systems presented in this article are based on modules of sound processing/generation that interact in feedback networks. The systems are exclusively algorithmic. This means that there is no coupling with the

real-world environment through electronic transducers, except for the adoption of gestural interfaces to provide to human performers an efficient channel to carry out parametric changes in order to drive the overall system's behavior.

These networks can be described as *time-variant* in two ways: 1) when the parameters of its modules are mapped to control-signals that are extracted from audio generated by other modules in the network, then changing its internal state in function of contextual conditions; 2) when the state of the system changes according to actions of a human performer. In this last scenario, two types of actions can be performed: changing modules' internal parameters, and changing the network topology.

The feedback networks described below were not planned to be adaptative in a strict sense, it means, their operational modules for audio processing/generation were not aimed to look for some sort of balance and homeostatic behavior with their environment (the network of modules), nor to achieve some goal, but were designed to provide autonomous and/or complex non-linear behaviors that could be used as resources for music creation.

3. Frequency modulation network

The first time-variant feedback network presented here was designed with eight modules of frequency modulation synthesis (FM), in which the modulation signal for each module is the scaled weighted sum of all modules' output signal, as showed in Equation 1.

$$E_n = \sum_{m=1}^8 A_m^n S_m$$

$$S_n = \cos(2\pi fp + En + \phi)$$

Equation 1: mathematical formalization of a networks' FM synthesis module.

Being E_n the input for a module n and S_n its output (for $n \in \mathbb{N}$; $n = [1 \dots 8]$). The amplitude coefficients for modulation signals are given by the matrix A_m^n in which n e m are indexes of the synthesis modules, fp is the carrier frequency in Hz and ϕ is the initial phase of each module oscillator.

Although most of the parametric settings for this system produce a broadband (almost white noise) sound that is of little interest to music creation, some specific configurations result in more varied and complex sounds. Because of its non-linear behavior, in which there is no correlation between sound features and parametric changes, and also due to minimal parametric changes can lead to completely different results, one cannot easily predict the sound generated by a new parametric configuration. Therefore, the task of finding sets of values that produce musically interesting sounds is a careful, almost handicraft one, done by trial and error.

Some of the well-tuned parametric settings can result in perceptually static sounds with differences in timbre, noisiness, pitchness, and register; others yield dynamic streams of sounds, some with noticeable periodic patterns in pitch and/or rhythms, others with more complex or random-like dynamics.

Despite being wide-ranging in parametric combinations and highly unpredictable in behavior, the system is deterministic and specific results can be retrieved if the same initial conditions were provided. Taking advantage of this feature, the approach to create a playable time-varying network was adapting a gestural interface that would allow fast reconnections of its modules, while the FM's parametric settings are kept constant. This method provides limited, yet broad, fields of generated sounds that can be permuted or interpolated.

A commercial MIDI interface with 64 buttons organized in a 8x8 matrix was adapted to be a gestural interface for switching connections in the network (Figure 1). The matrix rows represent modules output and columns their input. For the sake of simplicity, to reduce the number of possible connections and increase systems' playability, a constrain was added: each column accepts only one state, that is, each module can receive signal from only one module (with amplitude equals 1), while a module can send signal from up to eight modules. Input weighting is applied only during timed interpolations between matrix states, an implemented feature that can generate different emergent sound behaviors according to the interpolation time. This parameter, by its turn, can be changed through buttons in the rightmost column of the interface, which represent discrete time values in ascending order from bottom to top, within a customizable range. The top row of buttons was assigned the store and retrieve eight matrix state presets.

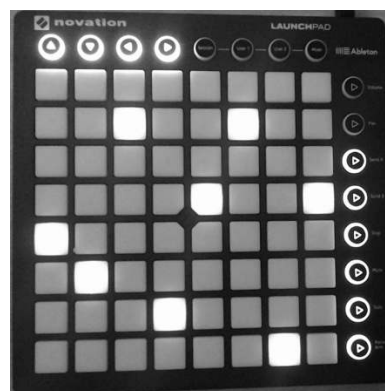


Figure 1: Physical interface adapted to switch network connections through a 8x8 matrix that represents connections between FM modules' inputs and outputs.

This interface allows a performer to quickly switch between connection patterns and intuitively explore the field of sonorities of a FM parametric setting.

Figure 2 shows the spectrogram of three different patterns of network connections for a single FM preset in which the carrier frequencies for module one to eight are: 39, 0, 21, 5, 57, 25.84, 16.01, 0.44; and the amplitude coefficients for their modulation signals are respectively: 41808, 741, 10617, 13680, 171, 4715, 526. The respective network topologies are represented by the matrices in Figure 3, in which 1 means a connection between two modules, and 0 their disconnection. Matrices A to C correspond to spectrograms in Figure 2 respectively from

top to bottom.

The upper spectrogram (related to matrix A) shows a sound texture composed of gliding simple tones in periodic movements across the frequency axis that take relatively long time spans (several minutes); the middle spectrogram (related to matrix B) displays alternations of complex spectrotemporal patterns lasting from some hundreds milliseconds to some seconds; the spectrogram at the bottom of Figure 2 (related to matrix C) displays a steady sound texture composed of almost stationary frequencies lasting about 100 milliseconds and mixed with a periodic noisy pulse train. Notice that small changes in the network result in widely different outputs, for instance, the single changed connection from matrix A to B, or the four changed connections from matrix B to C.

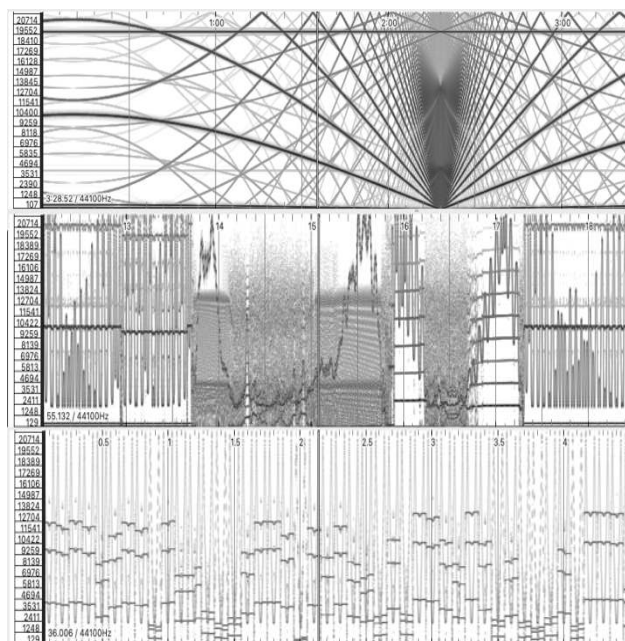


Figure 2: Spectrograms of three sound samples generated by the FM feedback network using a same FM parametric setting but with different network topologies.

A INPUT									
O U T P U T		I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈
	O ₁	0	1	0	0	0	0	0	0
	O ₂	0	0	1	0	0	1	0	1
	O ₃	0	0	0	0	1	0	1	0
	O ₄	1	0	0	1	0	0	0	0
	O ₅	0	0	0	0	0	0	0	0
	O ₆	0	0	0	0	0	0	0	0
	O ₇	0	0	0	1	0	0	0	0
	O ₈	0	0	0	0	0	0	0	0

B INPUT									
O U T P U T		I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈
	O ₁	0	1	0	0	0	0	0	0
	O ₂	0	0	1	0	0	1	0	1
	O ₃	0	0	0	0	1	0	1	0
	O ₄	1	0	0	0	0	0	0	0
	O ₅	0	0	0	0	0	0	0	0
	O ₆	0	0	0	0	0	0	0	0
	O ₇	0	0	0	1	0	0	0	0
	O ₈	0	0	0	0	0	0	0	0

C INPUT									
O U T P U T		I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	I ₈
	O ₁	0	1	0	0	0	0	0	0
	O ₂	0	0	1	0	1	0	1	0
	O ₃	0	0	0	0	0	0	0	0
	O ₄	1	0	0	0	0	0	0	0
	O ₅	0	0	0	0	0	0	0	0
	O ₆	0	0	0	0	0	1	0	1
	O ₇	0	0	0	1	0	0	0	0
	O ₈	0	0	0	0	0	0	0	0

Figure 3: Matrices representing the network topology for the sounds displayed in Figure 2.

In Figure 4, two spectrograms generated from segments of an improvised performance with the FM network are presented. The delimited blocks of contrasting spectral content in the upper spectrogram are the products of a performance with short interpolation time between matrix states (about 5 milliseconds). The bottom spectrogram shows the results of longer interpolated transitions (about 2 seconds) that produce gradual changes in spectral content, like saturation and filtering patterns, *glissandi* effects, or the emergence of a periodic pulse displayed on 1'55" in figure's timescale.

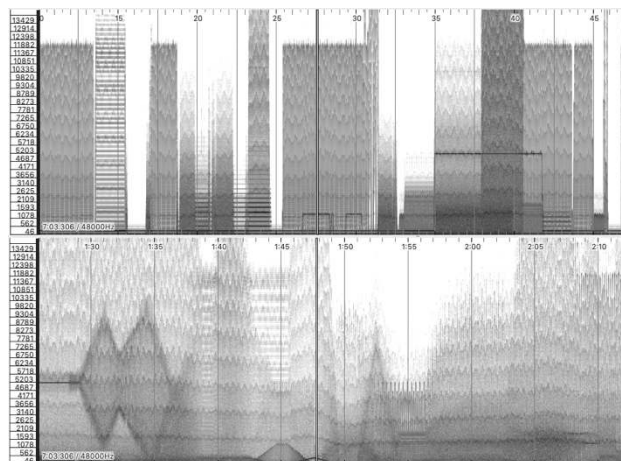


Figure 4: Two spectrograms of excerpts from a performance of the FM network using a small interpolation time (upper graph) and an interpolation time of a few seconds (lower graph)

Based on the author's practical experience, an in-depth use of this instrument requires a long-term work of at least three steps: first, an empirical research to discover FM presets that resultant sounds would fit the artist's aesthetic judgment; second, the exploration and memorization of matrix patterns, and the internalization of possible playing gestures in order to articulate meaningful sound changes in a musical context; and finally, using this acquired specific knowledge for music composition or improvisation.

4. Sample processing network

A second class of time-variant feedback system was designed based on transformations of pre-recorded audio samples. Unlike the system presented in the last section that has a single method of audio generation with direct signal feedback, this class of systems adopts a classical modular synthesizer architecture that allows for variability in the combination of audio processes, and it includes an audio feature extraction stage to generate control signals.

Figure 5 shows the general outline of the system. A feedback network is based on the combination of at least two modules. Each module is composed of three parts: a feature extraction stage, a sound generation stage, and an effects/post-processing stage.

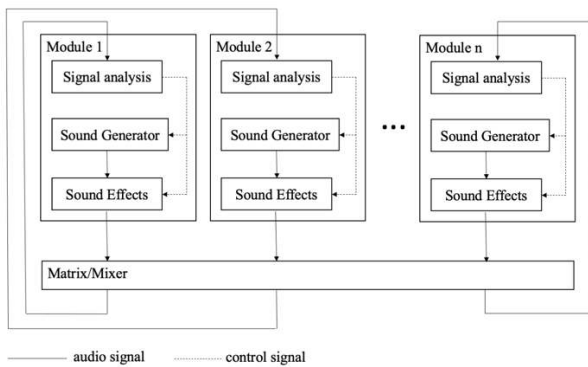


Figure 5: sample-based system architecture.

The feature extraction stage converts modules' input signal to control signals by applying temporal or spectral measurements on the weighted sum of all modules' output signals, such as RMS, spectral centroid, noisiness, spectral roll-off, among others described in the specialized literature [21]. The resulting signals are mapped to control parameters in the following stages.

Two methods of sample transformation were applied in the generation stage: playing a sample with variable speed (a scratch-like effect), and a time-stretch plus pitch-shifter effect based on granular synthesis [22]. In the effect/post-processing stage, a chain of audio effects is applied over the signal obtained from the generation stage, for example, variable state filters, modulations (amplitude modulation, ring modulation), panning, among others.

The instantiation of this general architecture in an unique configuration – i.e. the definition of audio processes and the mapping of control signals – was treated by the author as a matter of artistic choice that is related to the specific practical context for which it is intended, as some decisions (such as the selection of pre-recorded samples or the methods for sound generation and effects) have dramatic consequences in sound aesthetics. The choice of feature extraction algorithms, as well as the mapping of their outcome to sound synthesis/effects parameters, showed less influence on the overall aesthetic (the resultant sonority), but it was crucial for the temporal dynamics of the system, and consequently for the sound forms that emerge from system's interactions.

Figures 6 shows the interfaces of two modules implemented in the software Pure Data¹ that use a granular synthesizer for sound generation. In both of them, sample read position and transposition rate are parameters linked to control signals obtained from analysis of modules' input (which means the audio coming from other modules in the network). Effects were omitted from the module in Figure 6-A, while the one in Figure 6-B included a band-pass filter and a pan effect, both also linked to control signals.

In the networks composed with these two modules, the audio driven interaction mostly influenced the temporal dynamics of sound transformations that are operated by audio processing methods (unlike the FM network in which audio feedback defines the instantaneous

qualities of the sound synthesis, as well as its evolution over time). The result is a new sound texture formed by transformed chunks of the original samples.

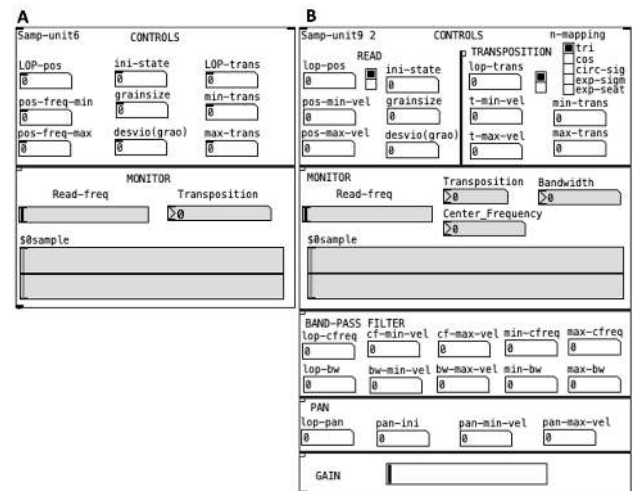


Figure 6: Interface of two sample-based modules implemented in Pure Data.

Figure 8 and 9 show three spectrograms generated from selections of audio recorded from a performance of a network composed of three modules of the type illustrated in Figure 6-A.

The modules were fed with sound samples which spectrograms are shown in Figure 7. The sound represented by the top spectrogram is a sequence of percussive pulses with resonance peaks ranging between 300Hz and 600Hz that were obtained from impacts of two plastic cups; the sound represented by the middle spectrogram consists of white noise bursts resulting from the manipulation of a masking tape; and the third sound sample, in the lower spectrogram, is a single cymbal strike with a drumstick that created a complex and inharmonic resonance pattern.



Figure 7: Spectrograms of three sound samples used to generate the examples of Figures 8 and 9.

For the network's output displayed in Figure 8, the transposition rates of the granulators were set to 1 (no

¹ <https://puredata.info/>

transposition) and their sample read position was modulated by a control signal. The result is a shuffle of time-stretched or time-compressed segments from the source samples which creates a continuous sound texture with fluctuation in rhythm and density, showing moments of greater or lesser activity and the prevalence of one or another module's output.

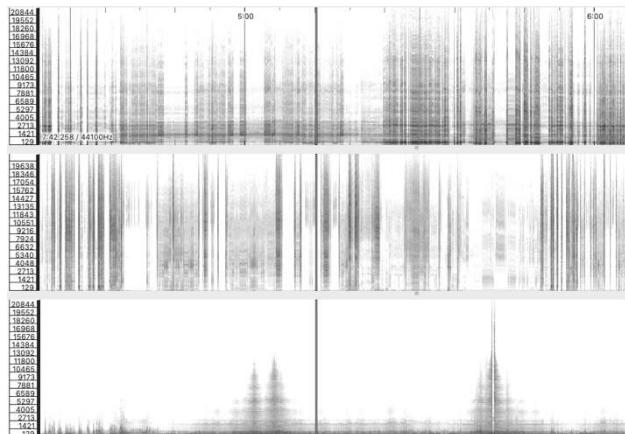


Figure 8: Spectrograms of the output produced by a sample-based feedback network composed of three modules of the type displayed in Figure 6-A fed with sound samples displayed in Figure 7, first example.

Figure 9 shows the result of a performance where both parameters in all modules (sample read position and transposition rate) were modulated by control signals. The modulation range for the transposition vary, being the broadest the third spectrogram (from up to bottom, related to the cymbal sample) and the narrower the first spectrogram (related to the plastic cups sample). Transposition modulations can be seen in the curves traced by spectral peaks or by more energetic spectral bands. As in Figure 8, chunks of the original samples are stretched or compressed and reorganized in new sound textures. A formal organization emerges with two processes of spectral saturation in the last spectrogram (cymbal sound) while the first two spectrograms (plastic cups and masking tape sounds) alternate between moments of high and low density of sound activity.

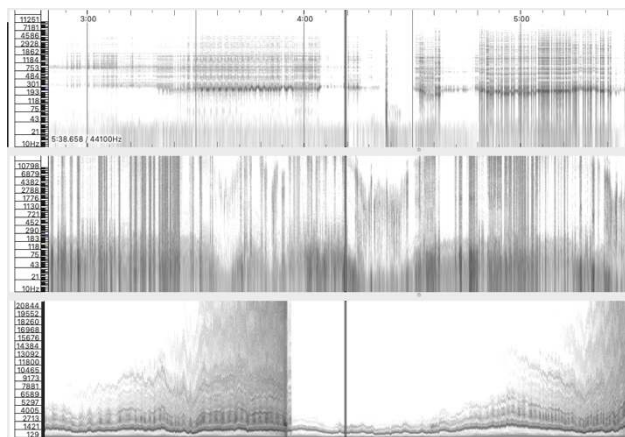


Figure 9: Spectrograms of the output

produced by a sample-based feedback network composed of three modules of the type displayed in Figure 6-A fed with sound samples displayed in Figure 7, second example.

Meaningful performative actions with this specific network relied mostly on changes in modules' parameters related to the audio generation/transformation processes, or related to the mapping and post-processing of control signals. Switching connections between modules showed less musical interest, as it affected the behavior of control signals on a small temporal scale without showing significant qualitative differences (it is noteworthy that this characteristic is related to this specific implementation and variations of the proposed architecture would result in different instrumental affordances).

5. Final considerations

The audio feedback network systems presented in this paper are contributions to the artistic research on autonomous generative music systems. They are snapshots of works in continuous development or transformation according to contextual and idiosyncratic needs that are characteristic of an artistic practice. From a technical point of view, future developments would be towards creating new specific implementations for the sample-based network architecture, or including post-processing/effects in the FM network.

References

- [1] C. P. Boner, and C. R. Boner. Behavior of Sound System Response Immediately Below Feedback. *Journal of the Audio Engineering Society*, 14(3):200–203, 1966.
- [2] T. Holmes. *Electronic and Experimental Music: Technology, Music and Culture*. (3rd Ed.). New York: Routledge, 2008.
- [3] N. Wiener. *Cybernetics; or Control and Communication in the Animal and the Machine*. New York: Wiley, 1948.
- [4] L. von Bertalanffy. *General System Theory: Foundations, Development, Applications*. New York: Braziller, 1968.
- [5] D. Sanfilippo and A. Valle. Feedback Systems: an Analytical Framework. *Computer Music Journal*, 37:2, pp. 12–27, 2013.
- [6] T. W. Patteson,. The Time of Roland Kayn's Cybernetic Music. *Travelling Time, Sonic Acts XIV*. Amsterdam: Sonic Acts Press, pp. 47–67, 2012.
- [7] A. Di Scipio. Sound Is the Interface: From Interactive to Ecosystemic Signal Processing. *Organised Sound*, 8(3):269–277, 2003.
- [8] A. Di Scipio. Listening to Yourself Through the Otherself: On background Noise Study and Other Works. *Organised Sound*, 16(2) 97–108, 2011.
- [9] A. Eldridge. *Collaborating with the Behaving Machine: Simple Adaptive Systems for Generative and Interactive Music*. Ph.D. thesis, School of Cognitive Science, University of Sussex, 2007.

- [10] P. A. Kollias, Ephemeron: Control over Self-Organised Music. In: *Proceedings of the Fifth Sound and Music Computing Conference*, pp. 138–146, 2008.
- [11] D. Sanfilippo. Time-variant infrastructures and dynamical adaptivity for higher degrees of complexity in autonomous music feedback systems: the Order from noise (2017) project. *Musica/Tecnologia*, 12(1):119-129, 2018.
- [12] S. Kim, G. Wakefield, and J. Nam. Augmenting environmental interaction in audio feedback systems. *Applied Sciences* 6, 5: 125, 2016.
- [13] H. Maturana and F. Varela. *Autopoiesis and Cognition. The realization of the living*. Dordrecht: Reidel Publishing Company, 1980.
- [14] A. Di Scipio and D. Sanfilippo. Defining Ecosystemic Agency in Live Performance. The Machine Milieu Project as Practice-Based Research. *Array: The Journal of International Computer Music Association*, p. 28-43, 2019.
- [15] S. Waters. Performance Ecosystems: Ecological Approaches to Musical Interaction. In: *Proceedings of the Electroacoustic Music Studies Network*, 2007.
- [16] P. A Kollias. The Self-Organized work of Music. *Organised Sound* 16(2): 192–199, Cambridge University Press, 2011.
- [17] P. A. Kollias Overviewing a Field of Self-Organising Music Interfaces: Autonomous, Distributed, Environmentally Aware, Feedback Systems. Proceedings of the 23rd Annual Conference on Intelligent User Interfaces – ACM IUI Workshops, Tokyo, 2018.
- [18] J. M. Chowning, The Synthesis of Complex Audio Spectra by Means of Frequency Modulation, *Journal of the Audio Engineering Society* 21(7): 526-534, 1973.
- [19] P. Maes, Modeling adaptive autonomous agents. *Artificial life*, 1:135-164, 1994.
- [20] M. Mitchell, . *Complexity: A guided tour*. Oxford University Press, 2009.
- [21] J. Bullock, *Implementing audio feature extraction in live electronic music*. PhD Thesis, Birmingham City University, Birmingham, 2008.
- [22] B. Truax. Discovering inner complexity: Time-shifting and transposition with a real-time granulation technique. *Computer Music Journal*, 18(2):38-48, 1994.

Lick the Toad: a web-based interface for collective sonification

Konstantinos Vasilakos ¹

¹Center for Advanced Studies in Music – Istanbul Technical University
Istanbul, Turkey

konstantinos.vasilakos@gmail.com

Abstract. *Lick the Toad is an ongoing project developed as a web based interface that runs in modern browsers. It provides a custom made platform to collect user data accessed from mobile devices, such as smartphones, tablets etc. The system offers a tool for interactive collective sonification aiding the idea of networked music performance. It can be used in various contexts, such as onsite installation, interactive compositional tool, or for the distribution of raw data for live coding performances. The system embeds neural network capabilities for prediction purposes by using user input and outputs/targets alike. The inputs and the targets of the training processes can be adapted according to the needs of the use making it a versatile component for creative practice. It is developed as an open-source project and it works currently as a NodeJS application with plans for future deployment on remote server to support remote communication and interaction amongst distant users.*

1. Introduction

The idea of network as a means for real time distribution is not anew in the field of electronic music making. It goes back to the pioneering Telharmonium by an American inventor Thaddeus Cahill [Manning 2004] and also influenced by other attempts, such as Soemmerring's Musical Telegraph, and Tivadar Puskás' Telefonhírmondó, and the Musical Telegraph by Elisha Grey [Crab 2013]. What these all had in common, was that they all attempted to enact the transmission of sound generated by custom made electric instruments via hitherto available broadcasting technology e.g., telegraphy and landlines. One can argue that these can be seen as primitive attempts for music distribution deploying networking, which presumably paved the way to a new dimension of musical practice and artistic expression. It allowed for the decentralization of the sonic information by transmitting the performance to various ends using radio and wired networks for telecommunications. One can see an arguably far fetched analogy with much of the current practices in the broad landscape of Sonic Arts, such as live coding and laptop ensembles which are using ad hoc networks as a means to establish real time communication and distribution of various data amongst each other during their improvisation, such as the Hub's initial attempts for this paradigm of computer music networked performance [Gresham-Lancaster 1998, Collins and Escriván Rincón 2011]. Of these, the Powerbooks Unplugged ¹ and the Birmingham Ensemble for Electroacoustic Research (BEER) ² [Wilson et al. 2014] are using custom made libraries for network communication, such as, the Republic (<https://github.com/musikinformatik/Republic>) and

¹See <http://wertlos.org/pbup/>

²See <http://www.beast.bham.ac.uk/offspring/beer/>

Utopia (<https://github.com/muellmusik/Utopia>) respectively. Both running in SuperCollider (<https://supercollider.github.io/>) programming language as external libraries, and are used to provide coders an interaction platform in order to communicate with each other as a means to thrust their live coding performances, making ad hoc networks inherent part of their research and musical practice.

In the wake of this performance paradigm, it helped to address some questions about *telepresence* and *liveness* via remote and distributed performers [Rohrhuber et al. 2007] using network communication. Besides the distribution of various data types through ad hoc local network some other examples using audio signals via the internet enacting remote communication include BEER's "Recalibrated" performance of a live coding piece entitled *Pea Stew* [BEER 2021], an adaptation of Nick Collins' *Pea Soup* [Collins 1974], and the *Symphony in Blue 2.0* by the Istanbul Coding Ensemble³, a live coding performance based in Kamran Ince's composition entitled *Symphony in Blue* (2012)⁴. While the former piece involves the feedback of audio signals amongst remote coders the latter uses piano signal for machine listening utilization both locally and remotely using SonoBus (<https://sonobus.net>), a low latency state of the art audio transmission software. Lastly, besides network being used as a medium to distribute audio information and/or data, some works have used directly its inherent elements, for example the Network Ensemble has used real time sound representations of raw packets from WiFi networks [Smith and Tacchini 2017], and Chris Chafe's music work entitled *Ping* [Chafe 2001] used the network's ping latency information to modulate sound synthesis parameters.

1.1. Sonification and auditory display

Sonification is the practice of rendering data to sound [Hermann et al. 2011, Worrall 2019]. Real time streams of data or static data sets can be used as raw material to build wave forms or create associations between data streams and sound synthesis parameters. This simple yet fertile approach for experimentation has been widely explored in the field of the Sonic Arts and digital arts alike. Examples include [de Campo 2007] and the deployment of astronomic and environmental readings to particle collisions data [CPNAS 2017, Hill et al. 2017, TEDx Talks 2013, Sosby 2019, Mohon 2021].

In addition, sonification practice has been the workhorse for both artistic and educational outreach. Previous works by the author are demonstrated in [Vasilakos 2019] and [Vasilakos et al. 2020]. Given the technical affordances sonification can be explored as web applications and can run in browsers of mobile devices, such as smartphones, tablets and whatnot. This provides a personal exploration tool and at the same time it offers decentralized environments⁵ which were previously unexplored due to the need of specialized background. Consequently, this opens up to non-technical backgrounds and the interested layman. Similarities in approach, includes IPSOS (<http://ipsos.web.cern.ch/>), a web-based application that runs on any modern browser and allows for the real-time sonification of data from the Large Hadron Collider, at CERN, in Switzerland.

³See <https://konvas.netlify.app/ice/>

⁴See <http://www.kamranince.com/html/compositions.php?id=189&page=>

⁵Decentralized environments refers to the standard model of laptop performances where the interaction relies on the single human computer relationship with the sonic generation.

1.2. Machine learning and the (Sonic) arts

Some examples that highlight using machine learning in music practice include Mar-ijje Baalman's "The Machine is Learning" [Baalman 2020], Sonami's and Fiebrink's [Fiebrink and Sonami 2020], and Sturm's work in the field [L. Sturm and Ben-Tal 2017]. Due to the development of machine learning modules in the browser, it allows for control without the need for specialized skills and interact with training and control of neural networks. This seems to be also observed by [Bernardo et al. 2020]. As Amershi et al. [Amershi et al. 2014] write,

IML enables even users with little or no machine-learning expertise [to] steer machine-learning behaviors through low-cost trial and error or focused experimentation with inputs and outputs.

2. Lick the Toad

Lick the Toad offers an interface to interact with other users that are connected in an ad hoc network and share their positions to one another and at the same time train a model of a neural network using this activity. The interface runs in mobile devices and modern web browsers using a URL as seen in fig: 1. For an overview of the system see this link (<https://tinyurl.com/2jk5bvyf>). It is an open source project and can be found at this link (<https://github.com/KonVas/lick-the-toad>).

Besides of the training data generation the users can also engage in collective prediction process and collaborate with each other to build a real time work e.g., providing it is placed as an interactive sound installation. Due to the variability of the data it can be safely speculated that it may allow for a wide range of unpredictability of the musical outcome. This assumes the unlimited associations between data streams and sound, enabled by mappings that one has to establish with the sound generation modules in advance. The project is using SuperCollider but any platform such as Max/MSP (<https://cycling74.com>) and Pure Data (<https://puredata.info>) can be employed using Open Sound Control (OSC) communication. Thus the platform allows for active engagement with the sound generation instead of experiencing a musical work that was created with a digital musical instrument (e.g., as seen broadly in NIME community), which often may be viewed by the audience as a black box requiring programming dexterity and/or technical background in music technology.

2.1. Collection and training

The interface involves a single object which transmits to all connected users continuously its X and Y positions as float values relative to the screen borders of the device, see fig:1. While someone is interacting with the rest of the connected users they can see a string of the ID, which represents each user currently interacting. The system is able to collect the data and render it to a normalized range of 0.0 - 1.0 and start the training.

In addition to the X and Y values of the object, the user is able to add a target/output value. A snippet of the collected data prior of normalization can be seen in the "data" JSON snippet. The ys is the output value, that is the targeted value for prediction here represented as *frequency*. Numerous iterations and similar data sets using this template are

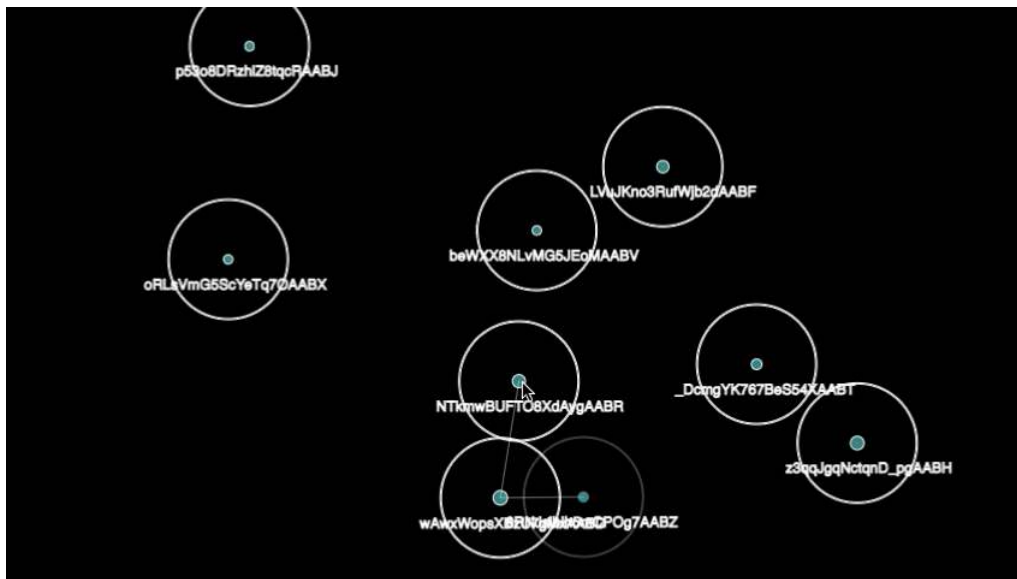


Figure 1. Interface for simultaneous user interaction.

stored in JSON format, the file can be saved locally in order to load the model and/or run the training process later. One can add other targets instead of *frequency* according to the needs providing that it follows the same format. The interface also creates visualizations of the receiving data using Berzier curves ⁶ as seen in fig:3.

```
{
  "data": [
    {
      "xs": {
        "x": 226.322645526962,
        "y": 84.91814510458188
      },
      "ys": {
        "frequency": 660
      }
    }
  ]
}
```

2.2. Training the model

The platform offers a minimal interface to start the various actions of the interface, such as collecting and training the model of the neural network. These actions are triggered using the native keyboard of the device. The neural network is implemented using the ML5 library (<https://learn.ml5js.org/#/reference/neural-network>), which offers a graph to monitor the training process as seen in fig:2.

It can be also monitored via the terminal window of the web page providing information about *loss*, *epochs* etc. of the training progress (see fig:4).

⁶<https://tinyurl.com/ykd6ywnu>

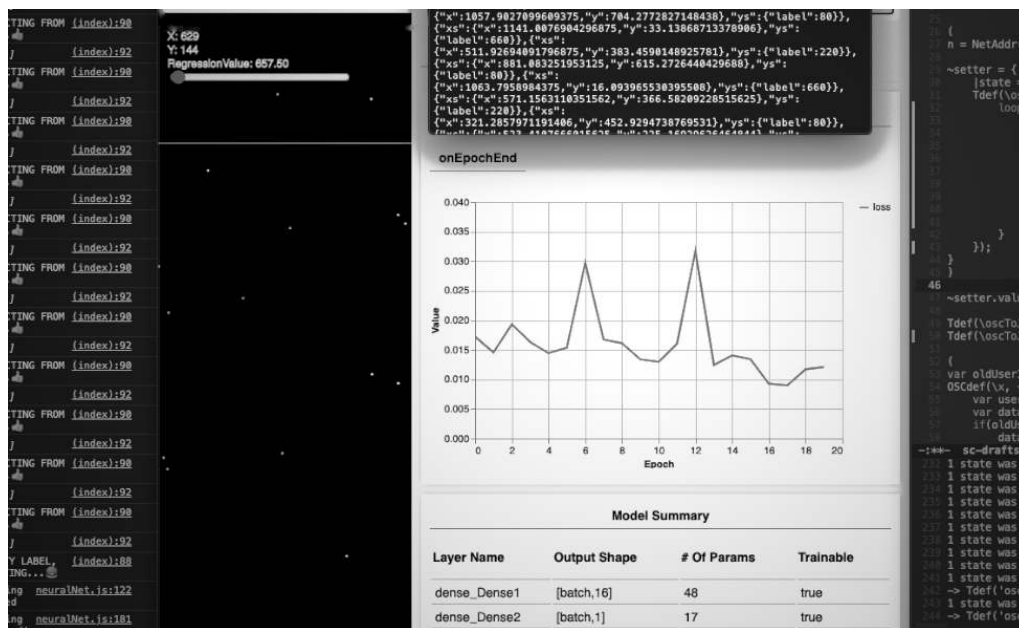


Figure 2. General overview of the interface.

2.3. Prediction and output

Once the system completes the training it starts generating prediction values based on the model, and triggering float values corresponding between the proximal points, for example, a user might enter some input data and declare it as *low*, *mid*, and *high*, their corresponding values (80, 220, 660) respectively; The output value will provide a continuous float number known as *regression value* ($=x$) between the targets, that is between *low* and *mid*, then this will be:

$$low \leq x \leq mid$$

The selection is based on a cursor according to its relative *X*, *Y* positions. This is controlled manually or randomly using a Perlin noise operator⁷ as shown in fig:5.

2.4. Mapping interaction

In addition to sending the regression value for sound mapping, the system also outputs the *X* and *Y* position of the cursor, that is, *X* and *Y* float numbers. For example, providing the cursor is between two points that are logged as *low* ($=80.0$) and *mid* ($=220.0$) the system will provide a value according to the proximity of these two and the same stream of the OSC message *X* and *Y* will be added. This format was found to be the most convenient for mapping the values to higher level musical parameters, such as frequency or duration to synthesise a sonic event.

The system can host diverse classification according to the nature of the input/output data, and thus generate a broad range of value range and mapping with sound sources accordingly. That allows the system to provide a versatile tool for experimentation in a sound synthesis context, instead of something that can be used only in a specific context and/or predetermined way which was devised by the author. For example, in live

⁷<https://p5js.org/reference/#/p5/noise>

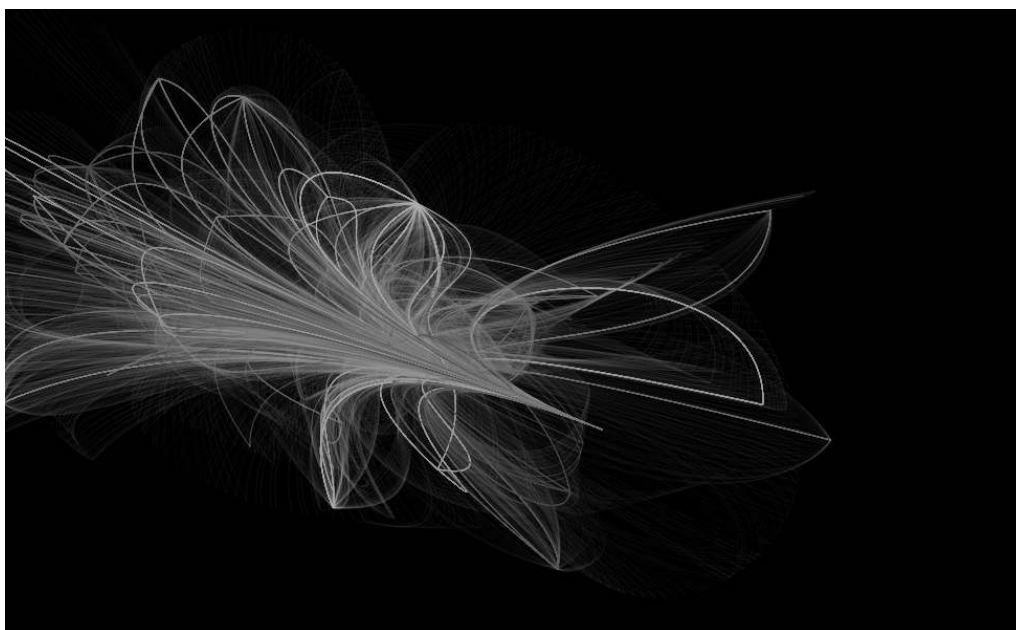


Figure 3. Visualization of user data using Bézier curves.

coding performances the system can acquire other data types to train the neural network. Successful tests using patterns to train the model generated from SuperCollider and used for live coding are also possible as seen in fig:6.

After the model is created it can be stored locally in the computer's hard disk and load again later for using it with other sound sources and software for musical creation.

3. Technical Details

TensorFlowJS (<https://www.tensorflow.org/js>), is a library for machine learning in the browser. ML5 is built on top of TensorFlow. It offers a higher level interface to implement machine learning in the browser including neural networks, pose recognition, and image prediction using pre-trained models offered by MobileNet (<https://github.com/tensorflow/models/tree/master/research/slim/nets/mobilenet>). More information about the library is found at this link (<https://ml5js.org/>). To provide real time data distribution amongst users the project is using Socket.io (<https://socket.io>), a library that is designed to provide to applications real time communications, such as event handling. To communicate with SuperCollider the project uses OSC communication implemented with OSC.js (<https://github.com/colinbdclark/osc.js>) library. NodeJS, the runtime of JavaScript also provides a local web server, which is built on top of the interface. This is also tested on a Raspberry Pi 3 Model B (<https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>). The visualizations are built in P5JS (<https://p5js.org/>), a link to the script can be found at this link (<https://github.com/KonVas/lick-the-toad>).

4. Creative Outcomes

Telematic art has been at the forefront of live coding practice and seems to provide a fertile solution to creating projects that focus on connecting remote user interactions, es-

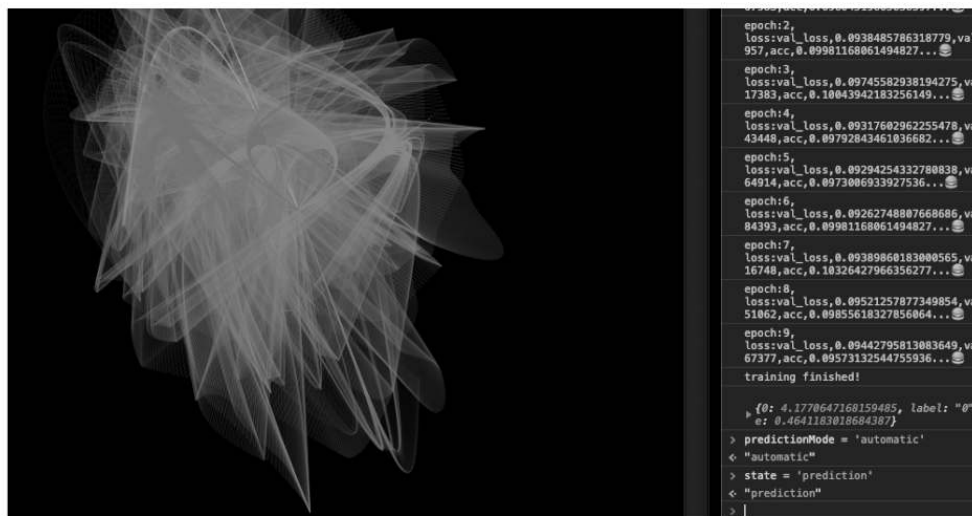


Figure 4. Training process and selection of prediction mode.

pecially in the wake of isolation and remoteness periods this may provide a useful tool for interaction and creative collaboration. While the project is in its initial stages, there are no performances that demonstrate the project's capabilities in such context. The system is a non-opinionated application that can be used in various ways that is, using data from logged users, or various streams generated by other applications and thus it can be presumed that it may be an interesting asset in the creative applications as follows.

4.1. Raw material for live coding

In the context of digital arts, live coding is the real time alteration of source code of running algorithms that generate sound or visuals [Collins et al. 2003, Nilson 2007, Zmólnig and Eckel 2007]. Live coding has been established as an emerging computer music paradigm with many enthusiasts using various tools and environments for improvising and tweaking source code in front of the audience. More information including tools and environments developed for live coding can be found at this link: (https://toplap.org/wiki/Main_Page). Previous works by the author in the field of live coding combining sonification practice includes [Vasilakos 2019, Vasilakos et al. 2020].

Besides the standard way of interacting with the native interface of the system it can be used to steer live coding performances. For example, providing decisions in the mapping between the incoming data and sound synthesis parameters. An exploration of this approach would include, generating the prediction values and streaming them to SuperCollider which it can adapt the data in additional functions that entail specific sounds to be triggered alike, e.g., *low*, *mid*, *high*, yet a very limited example for expressing the concept, as the targets can be adapted to much more detail and provide more sophisticated predictions based on the trained model. Then, the coder can improvise with the sound generation sources in SuperCollider in real time using all live coding techniques accordingly, (e.g., using JITLib in SC). During this performance one can explore live mapping strategies and its variants between the classification/prediction streams from the

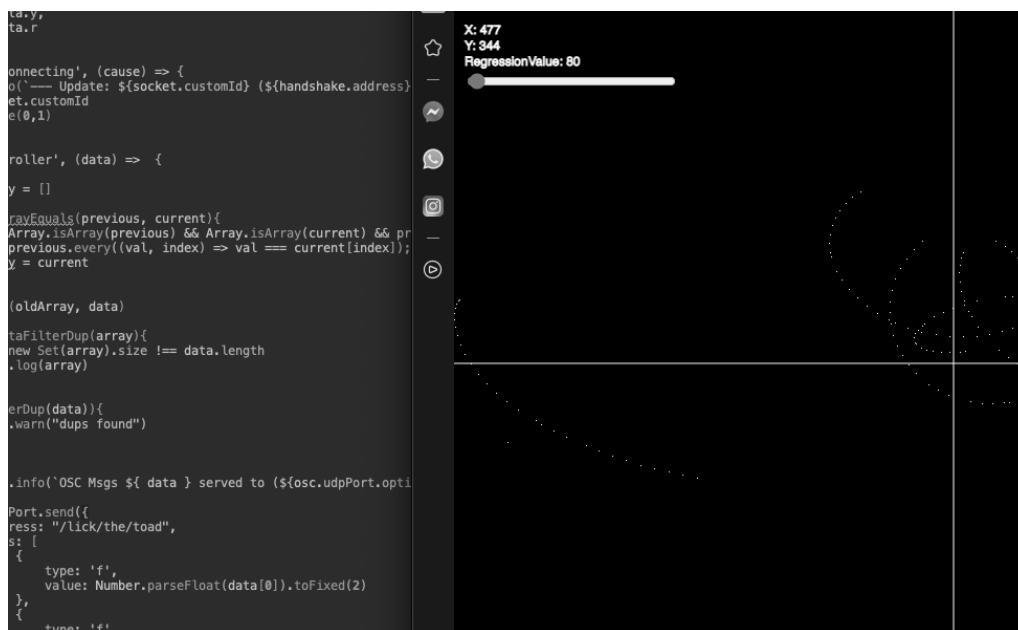


Figure 5. Selection of prediction using a cursor.

platform and the sound design algorithms generating an unpredictable musical outcome every time the system reiterates over the prediction data.

5. Future Directions

The application is at the stage of finalizing and running it via a local server in NodeJS. While it is an ongoing project, it is constantly adapted upon the requirements of projects. One of the future projects that the system may be used is with acoustic instrumentalists and train a model of the neural network using sensors for gestural information, and interpreted in various ways.

The primary goal of the project is to make the system accessible using a mobile device for online interactions. That means the users can interact from distant locations to train the system as well as interpret the data to create telematic sonification work. One possible way to implement this reciprocal interaction between training and sonifying amongst remote user(s) is to embed a remote server project developed by Serkan Sevilgen, a demo of this project can be seen at this link (https://www.youtube.com/watch?v=uP1_Jp-sfOc). For this it is required to be deployed and hosted on a remote server; this remains to be implemented in due course and finalization of other features will be implemented accordingly. Thus, next steps include the deployment of the platform as an online app which can host remote interactions amongst users across the internet and distribute data for various purposes, e.g., on site sound synthesis clients or live coders alike. Finally, further directions of this project will provide the combination of cosmetic and more eloquently appealing visualizations of the data, both in training and prediction modes. Finally, another future implementation which can be considered to expand on the creative outcomes of the system will provide the interaction between sound events and their visualization in real time, that is the sound events to be create real time graphics in another page of the interface.

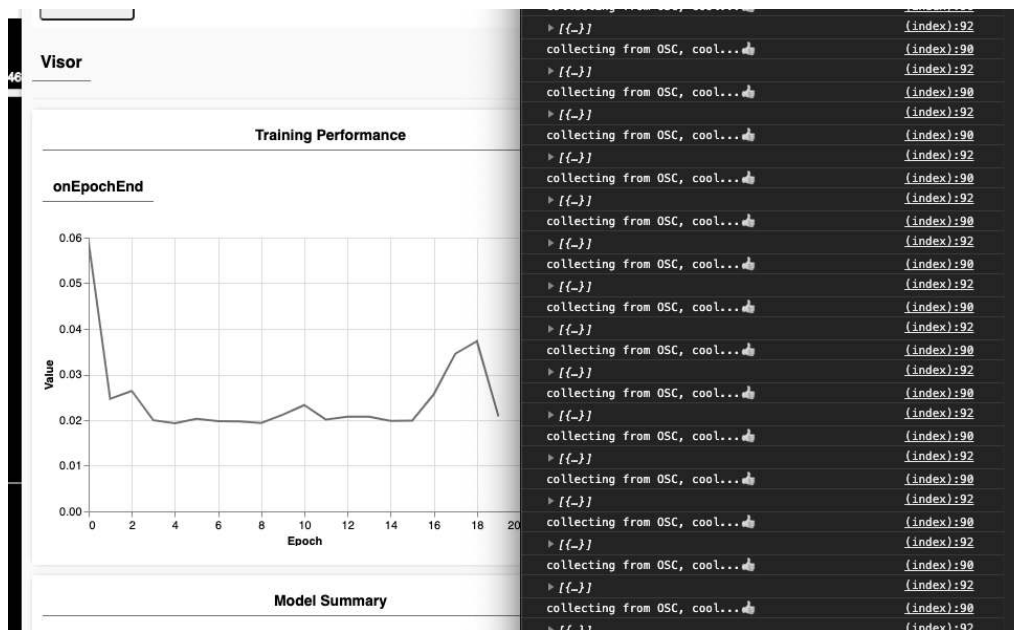


Figure 6. Collecting Open Sound Control (OSC) data generated in SuperCollider.

6. Conclusion

With the advent of machine learning technologies in the browser and its distributed collaborations it opens a brave new area for experimentation and musical research. Placing the audience in the forefront of the compositional epicenter instead of being a passive listener enables the collective interaction and real time communication amongst users and performers alike. This will provide for another level of exploration of creative nuances but also challenges for computer music practitioners and digital artists. Lick the Toad provides an interface to collect data from mobile devices and train a model to create further real time sonifications, which can be both triggered by the participants and/or used for live coding performances alike. So far, the system runs as a local environment in ad hoc network using NodeJS and SuperCollider for sonification purposes. The project has been proven to be quite flexible in terms of the collection of training data and thus it can be adapted to various projects, e.g., instruments and bystanders such as in the context of an on-site installation. Once the model is trained the interpretation can be mapped using OSC and SuperCollider as its sound engine. Further explorations, will allow for a fully decentralization of the data collection and run as a mobile application hosted and deployed on a remote server. This will allow for a wider interaction amongst users across the network.

References

- Amershi, S., Cakmak, M., Knox, W. B., and Kulesza, T. (2014). Power to the People: The Role of Humans in Interactive Machine Learning. *AI Magazine*, 35(4):105–120.
- Baalman, M. (2020). the machine is learning.
- BEER (2021). BEAST FEaST 2021: Concert 3.

- Bernardo, F., Zbyszyński, M., Grierson, M., and Fiebrink, R. (2020). Designing and Evaluating the Usability of a Machine Learning API for Rapid Prototyping Music Technology. *Frontiers in Artificial Intelligence*, 3:13.
- Chafe, C. (2001). Ping.
- Collins, N. (1974). Pea Soup.
- Collins, N. and Escriván Rincón, J. d. (2011). *The Cambridge companion to electronic music*. Cambridge University Press, Cambridge. OCLC: 781183853.
- Collins, N., McLEAN, A., Rohrhuber, J., and Ward, A. (2003). Live coding in laptop performance. *Organised Sound*, 8(3):321–330.
- CPNAS (2017). Mark Ballora, Seeing with Your Ears: Data Sonification.
- Crab, S. (2013). The ‘Telharmonium’ or ‘Dynamophone’ Thaddeus Cahill, USA 1897. <https://120years.net/the-telharmonium-thaddeus-cahill-usa-1897/>.
- de Campo, A. (2007). A DATA SONIFICATION DESIGN SPACE MAP. In *Proceedings of the 2nd International Workshop on Interactive Sonification, York, UK*, page 4, York, UK. ISon.
- Fiebrink, R. and Sonami, L. (2020). Reflections on Eight Years of Instrument Creation with Machine Learning. In Michon, R. and Schroeder, F., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 282–288, Birmingham, UK. Birmingham City University. ISSN: 2220-4806.
- Gresham-Lancaster, S. (1998). The Aesthetics and History of the Hub: The Effects of Changing Technology on Network Computer Music. *Leonardo Music Journal*, 8:39.
- Hermann, T., Hunt, A., and Neuhoff, J. G., editors (2011). *The sonification handbook*. Logos Verlag, Berlin. OCLC: ocn771999159.
- Hill, E., Cherston, J., Goldfarb, S., and Paradiso, J. (2017). ATLAS data sonification : a new interface for musical expression. In *Proceedings of 38th International Conference on High Energy Physics — PoS(ICHEP2016)*, page 1042, Chicago, USA. Sissa Medialab.
- L. Sturm, B. and Ben-Tal, O. (2017). Taking the Models back to Music Practice: Evaluating Generative Transcription Models built using Deep Learning. *Journal of Creative Music Systems*, 2(1):1–29.
- Manning, P. (2004). *Electronic and computer music*. Oxford University Press, Oxford ; New York, rev. and expanded ed edition.
- Mohon, L. (2021). Data Turned Into Sounds of Stars, Galaxies, Black Holes.
- Nilson, C. (2007). Live coding practice. In *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*, page 112, New York, New York. ACM Press.
- Rohrhuber, J., de Campo, A., Wieser, R., van Kampen, J.-K., Ho, E., and Hölzl, H. (2007). Purloined letters. In *Music in the Global Village Conference*, page 7, Műcsarnok Budapest. MV.

- Smith, O. and Tacchini, F. (2017). *Network Ensemble Selected Network Studies*. Number SF0010 in *The Strong of the Future*. Rizosfera, Via Guido de Ruggiero, 6 – 42123, Reggio Emilia – Italia.
- Sosby, M. (2019). Data and Music: What 50 Years of Exploring Our Moon Sounds Like. TEDx Talks (2013). Listening to data from the Large Hadron Collider Lily Asquith |TEDxZurich.
- Vasilakos, K. (2019). Emerging Dark Matter Through Diverse Sonification Practices. In *Proceedings of the International Music and Sciences Symposium*, pages 719–729, Istanbul. TMDK.
- Vasilakos, K. n., Wilson, S., McCauley, T., Yeung, T. W., Margetson, E., and Khosravi Mardakheh, M. (2020). Sonification of High Energy Physics Data Using Live Coding and Web Based Interfaces. In Michon, R. and Schroeder, F., editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 464–470, Birmingham, UK. Birmingham City University. ISSN: 2220-4806.
- Wilson, S., Lorway, N., Coull, R., Vasilakos, K., and Moyers, T. (2014). Free as in BEER: Some Explorations into Structured Improvisation Using Networked Live-Coding Systems. *Computer Music Journal*, 38(1):54–64.
- Worrall, D. (2019). *Sonification Design: From Data to Intelligible Soundfields*. Springer Publishing Company, Incorporated, 1st edition.
- Zmölning, I. and Eckel, G. (2007). Live coding: An overview. *International Computer Music Conference, ICMC 2007*, pages 295–298.

Mapping, Triggering, Scoring, and Procedural Paradigms of Machine Listening Application in Live-Electronics Compositions

Vinícius César Oliveira^{1*}, José Henrique Padovani¹²

¹Institute of Arts - State University of Campinas (UNICAMP)
Rua Elis Regina, 50 Cidade Universitária - Zeferino Vaz, CEP 13083-854, Campinas - SP

²LaPIS / CrIS – School of Music/Federal University of Minas Gerais
Av. Antônio Carlos, 6627 – 31270-901 Belo Horizonte, MG

oviniciuscesar@gmail.com, jhp@ufmg.br

Abstract. *Since the advent of real-time computer music environments, composers have increasingly incorporated DSP analysis, synthesis, and processing algorithms in their creative practices. Those processes became part of interactive systems that use real-time computational tools in musical compositions that explore diverse techniques to generate, spatialize, and process instrumental/vocal sounds. Parallel to the development of these tools and the expansion of DSP methods, new techniques focused on sound/musical information extraction became part of the tools available for music composition. In this context, this article discusses the creative use of Machine Listening and Musical Information Retrieval techniques applied in the composition of live-electronics works. By pointing out some practical applications and creative approaches, we aim to circumscribe, in a general way, the strategies for employing Machine Listening and Music Information Retrieval techniques observed in a set of live-electronics pieces, categorizing four compositional approaches: namely, mapping, triggering, scoring, and procedural paradigms of application of machine listening techniques in the context of live-electronics music compositions.*

1. Introduction

In the last two decades, the search for new approaches related to techniques aimed at Machine Listening (ML) and Music Information Retrieval (MIR) in the live-electronics repertoire has expanded considerably. The growing interest in exploring the possibilities that such tools offer to music creation resulted in artistic productions that encompass the most varied aesthetics currents, technical approaches, and compositional strategies.

In this article, we seek to outline remarkable characteristics of the compositional use of ML/MIR tools in live-electronics/interactive music combining instrumental/vocal sounds and DSP processes. By observing some compositional strategies that have been explored for composers in the context of ML/MIR and DSP techniques for the extraction of musical data, we aim to circumscribe a series of approaches by organizing them into compositional paradigms.

The article comprises two main parts. In the first one, we seek to situate ML/MIR as interdisciplinary re-

search fields, placing in context these areas and tools both in music researches/application as well as in a broader scope. In the second part, we seek to identify approaches to the use of ML/MIR processes in creative processes involving interactive systems, outlining four distinctive compositional paradigms in the field of interactive music: (1) *mapping*; (2) *triggering*; (3) *scoring*, (4) *procedural*.

2. ML and MIR as Interdisciplinary Research Fields

As interdisciplinary research fields, ML and MIR involve several areas, comprehending computer science, electrical engineering, acoustics, psychoacoustics, music theory, among others. In general, researches and applications in these fields involve the investigation and development of algorithms and DSP models that allow for the estimation of distinctive sonic/musical attributes from audio signals/streams and the extraction of correlated quantitative data through audio descriptors (including, possibly, machine learning and statistical methods).

By employing different DSP methods and psychoacoustic models, ML/MIR techniques retrieve quantitative data that, through a correlation with some semantic characteristics of sound, may give relevant clues about qualitative and perceptual elements of sound and music. These techniques allow for data retrieving from audio signals that can be useful to outline both more traditional music concepts (pitch, tempo, key, and intensity, for instance) as well as psychoacoustic-based parameters (like harmonicity/inharmonicity, brightness, roughness, loudness, among others). To the extraction of these features, many higher-level ML/MIR algorithms rely on cognitive and psychoacoustic models, seeking to simulate organs, mechanisms, and specificities of human auditory perception [1, 2].

According to [1], most of these algorithms are based on hypothetical modeling of human auditory perception and seeks to mimic psychoacoustic phenomena. There are several research areas aimed at the study of the operation and behavior of the human auditory system. For instance, the Auditory Scene Analysis (ASA) [3] studies how our cognition and auditory perception are capable to decompose and separate overlapping sounds and, consequently, identify them as their sound sources. In this sense, recent studies in the ASA field are focused on computa-

*Supported by CAPES.

tional modeling of these methods [4, 5], thus giving rise to Computational Auditory Scene Analysis (CASA), a new research field with several interests, motivations, and goals in common with ML and MIR fields.

In the context of computer-music researches, the term “machine listening” was firstly used to refer to the extraction of symbolic music parameters associated with the extensive use of MIDI protocol and its employment in the design of interactive music systems [6]. Later, the evolution of MIR, as a study field, was largely related to the development methods for analyzing and extracting sound attributes directly from digital audio, [7], having as initial motivation the implementation of indexing mechanisms for digital music files; content-based music search, retrieval, and recommendation systems; and interfaces that allowed easy access to massive online collections.

In the present context of technical and creative researches on the real-time computation of music and sound, both ML and MIR refer to processes of extracting quantitative data from audio streams and/or files. Specially since the second half of the '90s, within the spread of personal computers, and consequently with the emergence of musical computing environments — such as *Pure Data*, *Max/MSP*, and *Supercollider* —, the implementation of digital signal processing techniques was facilitated, allowing the creative exploration of real-time ML/MIR based techniques.

3. Compositional Paradigms

To understand the technical realizations, compositional approaches, and poetic conceptions that composers employ in the creative exploration of the set of tools of our interest, we propose an arbitrary organization of the different ways of thinking about its use into four groups, namely: (1) *mapping*; (2) *trigger*; (3) *scoring*; (4) *procedural*. We identified these paradigms regarding strategies of the conception of the algorithms, the organization and treatment of the extracted data, and the approximations on the creative use of these techniques to create the most diverse computational processes within the scope of live-electronic music.

We emphasize that the pieces discussed in this text are not meant to be representative of any aprioristic paradigm, which would precede them as a model, category, or aesthetic group. The identification of these paradigms is consciously arbitrary, built upon analytical interpretations. The objective, however, is not the categorization of these paradigms but to understand different poetic approaches regarding the use of specific tools in music creation. The idea is to identify poetical stances with regard to the creative application of the tools, aiming less at the segregation of particular aesthetics or compositional schools and more to comprehend how the insertion of these new tools into heterogeneous poetic practices takes place.

3.1. Mapping

We classify as *mapping paradigm* the creative applications of MIR/ML tools that rely on correlations and transformations between input parameters and the algorithms of

sound transformation/generation. It is strongly related to the processes of “liaison strategies between the outputs of the gestural controller and the input controls of the sound generation”[8] units in the context of the design of New Digital Instruments, allowing for building more simplistic or more complex relationships between these values and the synthesis algorithms they control. This correlation happens through a mediation layer, which is in charge of defining the paths that the input information stream will take, and the function that they will acquire within the architecture of the synthesis processes [9]. In short, mapping operates on transferring information and converting it into parameters in order to control computational actions and processes.

Mapping is a characteristic process of digital luthiery, i.e., the design of digital/expanded instruments and digital music interfaces [8, 10, 11]. While in acoustic instruments, the interface is linked directly to the sound production mechanism — the violin strings are an integral part of both the control and the sound generator mechanisms —, in digital instruments, in general, the interface is separated from the sound production device: the relation between them needs to be defined, what is done through mapping processes[12].

In general, mapping processes are conceived through pre-defined strategies, which correlate in straightway a set of data to a series of parameters — e.g., the roughness rate extracted from a signal can control the grain size of granular synthesis. Therefore, these correlations can be divided into four categories: (1) one input data controls only one parameter (one-to-one); (2) one input data control more than one parameter simultaneously (one-to-many); (3) more than one input data controls only one parameter (many-to-one); (4) a combination of all three categories, in which more than one input data controls more than one parameters (many-to-many) [13].

A practical example of mapping application can be observed in the chamber opera, *PS. I will be home soon!* (2012), for six instruments, four voices, and live-electronics, by Anna Einarsson. The libretto was written by Maria Sundqvist and is based on the life of the Swedish sailor, Calle Pettersson covered in the book *Kung Kalle av Kurrekurreduttön* [14].

Considering the technical and musical aspects, Anna Einarsson developed tools and procedures aiming at musical/spectral information extraction from the singing voice, mapping these parameters to control the interactive processes of the piece. According to the composer, the idea since the beginning of the compositional process was to develop mechanisms of dynamic control of the electronics, without requiring physical interfaces or sensors to be coupled to the performers [15].

In collaboration with the researcher Anders Friberg, Einarsson has implemented in *Ps. I will be home soon!* an extraction method for retrieving attributes related to the vibrato range produced by the singing voice. Such a method relies on an algorithm called CUEX (Cue-

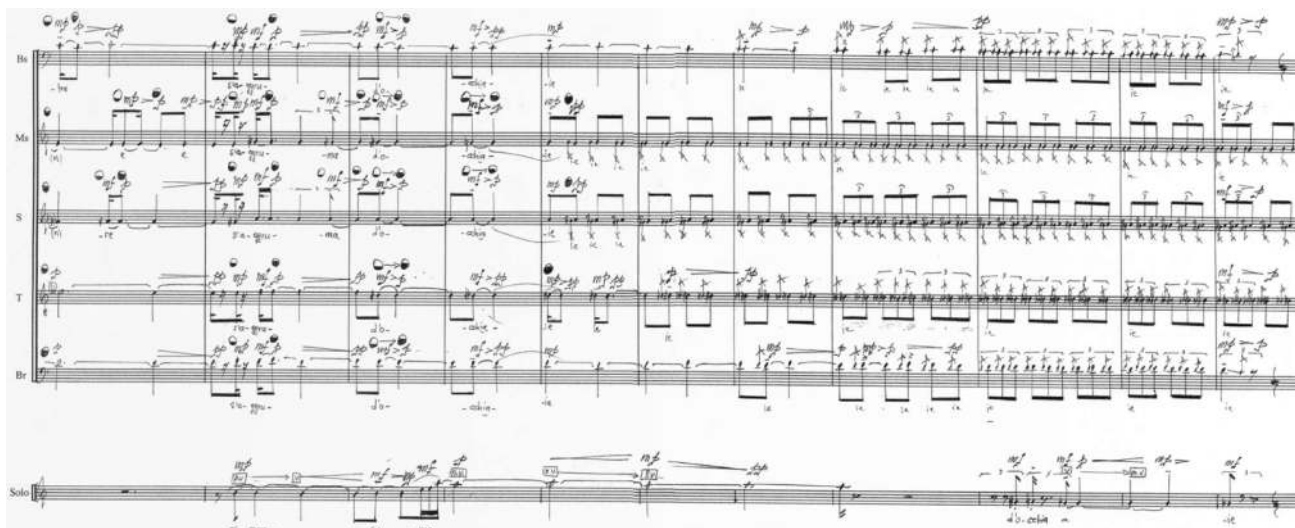


Figure 1: Excerpt with vowel tremolos referring to bars 122 to 130 of *Da una crepa*, by Marta Gentilucci

extractor), developed by Friberg in MatLab and, posteriorly, ported for *Pure Data* [16]. The extracted information from the voice is mapped and used to control four polyphonic synthesizers, each of them containing resonant filters and oscillator banks, which are excited by the noise signals [17].

The extraction process can be divided into five steps: (1) fundamental frequency detection and signal amplitude measurement through the Yin algorithm [18]; (2) creation of a gate using a specific pitch range and an amplitude threshold (only the pitches which fall within this limit will pass to the analysis); (3) the signal is divided into two parts, the first part is filtered by a low-pass filter and the second by a medium filter with a window size equals to a vibrato cycle — considering a vibrato rate equals to 6hz; (4) the two parts are subtracted in a way that the resulting signal will be zero when there is no vibrato; (5) the average of the absolute signal value is calculated through a medium filter with the double size of a vibrato cycle (thus the signal resulting will be proportional to the extent of a vibrato given in cents) [16, 17].

In *Ps. I will be home soon!* the mapping process is related to the notion of *metaphoric mappings*, such as described by [19]. What is transferred, in this process, are not the attributes themselves, but the relationships between the source domain (origin) and the target domain (destination). According to [17], in this piece, there is a relationship of forces employed to the mapping method. The density expresses the force in the target domain, which represents a dynamic structure of grouping of pitches, and the vibrato is the expression of the force in the source domain, i.e., the singing voice. Thus, the greater the presence of vibrato, the denser the grouping of pitches resulting from the synthesis process.

Another piece that illustrates very well the employment of what we call here mapping paradigm, is *Da una crepa* (2011-2012), for choir, soprano, ensemble, and live-electronics, by Marta Gentilucci (Figure 1). Some of

the central compositional aspects of *Da una crepa* arise from Gentilucci's desire to explore rhythmic structures, both at the macro-level — controlling and developing complex rhythmic structures throughout the composition — and at the micro-level — relationships between the internal temporal structures of the sound, and the possibility of using these structures as a control interface to generate and/or manipulate other sounds.

In this context, there arises the idea of correlating the rhythmic structures generated by vowel tremolos extracted from the singing voice — i.e., the repeated alternation of vowels over the same note, for example, “a-e-a-e” — to parameters that could manipulate pitches in electronic processes. In other words, the alternation of vowels would be applied to a given sound to control the amount of variation in the transposition of that same sound. In short, the composer intended to design a process that could modulate the pitch of a tenuto (sustained) sound through the rhythmic structure generated by the articulation of the voice.

Thus, the process of extracting and mapping the attributes used in *Da una crepa* is carried out in three steps. Throughout the piece, certain excerpts are recorded and saved in buffers. Then, these segments are analyzed with the help of audio descriptors (a combination of spectral deviation, odd to even second tristimulus, and total energy to extract the vowel tremolo [20]), and the resulting data of this analysis, being stored in a [coll] object¹. Finally, the stored data maps the transposition of two modules of pitch shifter and a harmonizer.

3.2. Triggering

The *triggering paradigm* is related to the ML/MIR employment in the conception of systems and algorithms aimed at sound/musical event detection. This strategy deals with procedures that rely on logical processes. In

¹Max/MSP object that allows the storage, organization, editing, and retrieval of messages

Figure 2 shows two excerpts of musical notation for the piece 'Decostructing Dowland'. The top excerpt (red border) covers bars 11 and 12. It features a guitar (Gtr.) staff with a treble clef and a computer (cmp) staff with a 4/4 time signature. The guitar part includes a 'pull-off Blue-Tack' and a 'slide RH' with fingerings 1-6. The computer part has a trigger labeled 'T2 (Activate auto trigger pre-prep sound)'. The bottom excerpt (blue border) covers bars 74 to 76. It also features a guitar (Gtr.) staff with a treble clef and a computer (cmp) staff with a 4/4 time signature. The guitar part includes a tempo marking of 50 and a note: 'NOTE: fingered pitch is not the sounding pitch'. The computer part has triggers labeled 'T29 (Activate auto trigger pre-prep sound)', 'T30 (Activate auto trigger pre-prep sound)', 'T31 (Activate auto trigger pre-prep sound)', and 'T32 (Manual trigger pre-prep sound, 5 seconds)'.

Figure 2: Bar 12 (red) and bars 74 to 76 (blue) of *Decostructing Dowland*

other words, algorithms analyze the digital signal looking for specific characteristics, which can describe certain events, such as pizzicato; dynamic variation; trills; vibrato, among others. Upon the detection of such events, the algorithm may trigger some electronic process.

The triggering paradigm is widely used within the scope of *live-electronics*, as well as in the design of *interactive musical systems* and autonomous agents [1], as a direct interaction strategy between the machine and the performer. By “listening” to the instrumentalist and detecting the presence of pre-defined events, the computer triggers processes, thus establishing a direct relationship between the instrumental gestural actions and the response of the machine to these impulses.

Decostructing Dowland (2009), for acoustic guitar and live-electronics, by Natasha Barret, illustrates in a simple and didactic way the practical application of the triggering paradigm (Figure 2). With the desire to unite ancient music with current musical practice, Barret takes, as a starting point, the three galliards for lute solo published in the first book of songs by John Dowland.

In this piece, the triggering paradigm occurs through the application of an attack detector to automatically and accurately trigger small sound samples previously conceived in the studio and saved in buffers. To realize this, Barret uses the `[bonk~]` object, implemented by Miller Puckette [21], for *Pure Data* and *Max/MSP*.

The `[bonk~]` works in the audio signal analysis to detect attacks, through the search for sudden changes in the energy envelope, which presents specific characteristics. The detection process is applied in bar 12 and in the excerpt that goes from bar 74 to bar 76 of *Decostructing Dowland*. In these two excerpts, the `[bonk~]` responds to Bartók pizzicatos (measure 12), and the attacks produced by the artificial harmonics (measures 74 to 76), triggering the pre-recorded sound samples. Although such excerpts have relatively simple rhythmic structures, the automatic attack detection strategy results in more precise synchrony between the action of the instrumentalist and the

sample triggering. This provides an efficient amalgamation between acoustic instrument sound and the pre-recorded samples so that the triggered sounds work as a spectral extension of the acoustic guitar resonance.

3.3. Scoring

The *scoring paradigm* is a way of thinking and conceiving the creative use of ML/MIR tools that broadly encompasses procedures, technical solutions, and compositional approaches aimed at the implementation of score following systems — i.e., methods aimed at tracking the musical performance (audio stream) and its synchronization to a score. Score-following processes rely on symbolic representation of the musical events (score) and on musical data extracted by ML/MIR algorithms. The score and the ML data are compared, with different strategies, allowing for the estimation of the probability that the instrumentalist is performing a specific section or musical event.

The first implementations of score following methods depended on the retrieval of symbolic information, strongly dependent on the extensive use of MIDI-based instruments in interactive music systems. Since the emergence of real-time DSP-based ML/MIR algorithms, score-following systems have started to get to recognize musical events of musical scores by analyzing audio streams. These new algorithms explore psychoacoustic, stochastic, and machine learning models and methods, relying on probabilistic heuristic methods and on diverse descriptor-based ML/MIR processes. These tools allow for the prediction of the performer’s tempo, yielding a variable metric variable related to the instrumentalist’s agogic — such as happens with *Antescofo*² [22].

By enabling the possibility of determining and prescribing with some degree of precision the actions that should be performed by a computer, synchronized to a live musician, the conception of morphologies resulting from computational processes begins to operate in the same domain as the instrumental writing process. Since musical notation has a symbolic and abstract character, it al-

²*Antescofo* is a modular score following system developed by Arshia Cont, which combines machine listening and machine learning methods

lows a musical gesture to be graphically represented and recorded on media — in this case, the score. Similarly, score-following-based processes bring close together live-electronics composition approaches and practices traditionally explored in compositions for acoustic instruments and fixed electronic media. By using score-based tools in live-electronics, composers may conceptualize sound morphologies that arise from the interaction between instrumental sounds and the computational processes [23].

One of the first pieces to creatively explore this set of strategies that we classify here as the *scoring paradigm* is *Jupiter* (1987), for flute and live-electronics, by Philippe Manoury. This is the first piece in a cycle called *Sonus ex Machina* composed in collaboration with Miller Puckette between 1987 and 1991, and stands out in the live-electronics repertoire, as it established new paradigms regarding the use of ML and interactive music systems — particularly, the synchronization and interaction strategies between electronic processes and instrumental performance [24, p. 147].

The score following mechanism implemented in *Jupiter* uses the `[scofo~]` object, capable of reading the MIDI pitches prescribed from a text file (accompaniment score) and comparing them with those extracted from the flute signal. For each note contained in the accompaniment score `[scofo~]` assigns a “theory”. The theory is responsible for describing the strength of the evidence that the most recent pitch played by the flute matches the most recent pitch read from the accompaniment score. If they match, the `[scofo~]` returns the number of the event linked to it.

If the pitch extracted from the live flute matches the pitch prescribed in the accompaniment score — which may or may not be linked to an event —, a control message contained in `[qlist]` footnote *Pure Data* object that reads text files containing time-stamped messages, which can be sequenced automatically or manually is sent to the transformation and synthesis modules, and a process is triggered. If there is no event linked to that note, the system will only receive the pitch and continue following the flow of data extracted from the flute. One of the main consequences of the use of *score following* in *Jupiter* reflects on the choice of the composer to represent the structures and sound/musical morphologies emerging from the flute-computer interaction through musical notation (representation of pitches, durations, and dynamics). Since the parameters of computational processes are precisely prescribed in `[qlist]` it is possible to predict the approximate behavior they will present when triggered, enabling the representation of these parameters using traditional notation.

This process of notation of electroacoustic events in the musical score converges with the notion of *Solfège of models*. This concept consists of knowledge, intellectual and cognitive skills acquired by the composer when designing certain processes, allowing to control and master the musical structures resulting from generative models and to associate graphic/textual representations of cer-

tain *software* and musical systems and the final musical result [25]. The use of *score following* linked to the idea of *Solfège of models* allowed the composer, from previous knowledge of the nature and parameters of the processes of sound transformation/generation involved in the piece, to glimpse their behavior when integrated with the flute gestures. This leads us to believe that an important step in the compositional process of *Jupiter* — and that seems to be one of the most present characteristics in musical writing of Manoury — concerns, precisely, the conception of the electronic part in an approximate way of the instrumental composition — ie, imagining musical structures and morphologies resulting from the amalgamation of computational processes and instrumental gestures, and reducing their sonic complexity to a symbolic abstraction that could be represented by traditional notation. The figure 3 shows the beginning of the section **ID**, of *Jupiter*, which shows the representation of the morphology resulting from the interaction between the flute (lower stave) and four modules of transposition of heights associated with a reverb (upper staves).

The desire to bring the concept of live-electronics computational processes closer to the notational dimension of instrumental writing seems to us to be a central issue that permeates the compositional processes that explore the scoring paradigm, as shown by some aspects of the compositional process of *Jupiter*. This becomes even more accentuated when we observe some compositional thoughts used in pieces that explore the use of the *Antescofo* as a tool that acts in the integration between these two environments.

An example of a compositional strategy that employs the *Antescofo* can be seen in *Dispersion de trajectoires* (2014), for baritone saxophone and live-electronics, by José Miguel Fernández. In this piece, *Antescofo* plays a central role in the process chain, as it is responsible for connecting, monitoring, and coordinating processes in different devices and computing environments. Like happens in *Jupiter*, there is in *Dispersion de trajectoires* a compositional thought that to some extent implies the use of some a media or technological apparatus aimed at the conception and registration of sound events in parallel with the instrumental score — again in an attempt to centralize the conception of computational/interactive processes and bring them into the realm of instrumental writing. This is evidenced by the architecture of the technical set created by the composer, which gives the *Antescofo* a leading role in coordinating processes throughout the piece.

As a technical strategy to work with instrumental writing and the conception of interactive processes in a centralized way, Fernández developed a library called *Antescollider*. This library integrates the *Antescofo* with the *SuperCollider*, allowing the conception of a centralized electronic score. The audio processing information is distributed to the *SuperCollider* servers using the Open Sound Control (OSC) protocol under the supervision and coordination of a score/script conceived in the *Antescofo* language. With *Antescollider* the audio processing can



Figure 3: The beginning of section Id of *Jupiter*

be distributed across multiple SuperCollider servers, thus maximizing CPU efficiency [26]

According to Fernández, the centralized score is relevant to the execution of the performance, since this model of employment of the *Antescofo* and the *Antescollider* creates support that makes it possible to explicitly write instrument-computer interactions. This setup also favors a dynamic approach to managing audio processing and synthesis — audio chains in SuperCollider can be easily created and destroyed dynamically, in direct response to the actions of the *Antescofo*.

3.4. Procedural

We classify as *procedural paradigm* every process that makes use of ML/MIR techniques to design algorithmic and compositional procedures implemented to operate in real-time. These procedures can be defined as a finite sequence of rules, reasoning, or operations that, once applied step-by-step to a data set, are capable of engendering structures as a final product.

Observing through the prism of the practice of Western instrumental and vocal writing, it can be seen that the notion of algorithm is already present in compositional procedures beyond the insertion of the computer in creative practice. The rules and formulae that guide contrapuntal writing or even the composition of dodecaphonic series, for example, are very close to concepts behind the definition of *algorithm*.

This set of compositional rules presupposes a particular formalization, in other words, a process of identifying a problem and choosing a solution. This formalization does not exempt the composer from searching for individualized solutions aimed at achieving specific aesthetic results. Although the need for non-rational thought coexists, such procedures aim at solving musical impasses through a set of logical steps, which demand the implementation of a congruent and well-formalized conception.

Therefore, the approximation between algorithms and compositional procedures is valid when they predict a symbolic and quantitative representation of music, and this is submitted to logical operations. That is, the procedures are defined by a formalization that demands the elaboration and definition of mathematical representations and automated operations on these representations. [27].

Before the integration of these techniques into musical thinking, compositional procedures — such as structural manipulations; rhythmic; melodic, and harmonic, as well as those aimed at organizing and transforming the sound material from the perspective of timbre and spectral behavior — took place outside the time of the performance, requiring prior elaboration and calculation to be able to materialize. With the advent of live-electronics, and above all, with the instantaneity and interactivity brought by ML/MIR tools, such procedures became part of the musical performance itself [28], being able to adapt to the most different musical situations, to bring the transformation/sound synthesis mechanisms closer to autonomous agents.

We can illustrate the use of the procedural paradigm through *Tensio* (2010), for string quartet and live-electronics, by Philippe Manoury (Figure 4). The *procedural paradigm* can be identified, in this piece, through the conception of an algorithm that acts just like a compositional procedure, engendering harmonic and melodic structures. The procedure relies on an additive synthesis process developed by Miller Puckette in 2006, called *3F synthesis* [29]. The *3F synthesis*, clearly influenced by the parallel frequency bands generated by modulatory processes such as FM synthesis and ring modulation, calculates a series of frequencies to be used in additive processes by manipulating input frequency values detected in real-time by ML processes/objects. Each of these three frequencies is subjected to all the possibilities of adding and subtracting each other so that it results in a wide range of frequencies. The generated values are then organized according to a probability index, which drives a stochastic filter of these generated parameters.

In a spectrum composed mainly of eight partials, if the probability index is equal to 1, the algorithm will select in ascending order the first eight calculated frequencies. If, on the other hand, the probability index decreases, the algorithm will select frequencies randomly farther apart from each other — which would, potentially, have more complex harmonic ratios. Thus, a proportional relationship between probability index and degree of harmonicity is established.

In *Tensio*, another compositional process that implies the employment of the *procedural paradigm* is the retrieval of musical information related to instrumental per-

Figure 4: Excerpt from *Tensio* that shows the interaction between the string quartet (lower staves) and the 3F synthesis (upper stave)

formance to create melodic structures driven by a Markov chain. Manoury uses pitches and durations values retrieved by real-time analysis algorithms to create a transition matrix capable of generating a sequence of pitches and durations based on the probability of appearance of each one of them after each other. In this case, this succession of pitches generated by the Markovian process is mapped to the first of the three frequencies used in the calculation of *3F synthesis* procedure, which will ultimately engender melodic structures.

The algorithm responsible for carrying out the Markovian process has some control parameters that influence the characteristics that the resulting melodic structures will present: (1) tempo of the melodic lines; (2) interval contractions and expansions; (3) pitch duration control (the higher the value, the greater the uniformity of durations); (4) transposition control; (5) minimum and maximum amplitude range; (6) interval multiplier that allows overlapping streams.

4. Conclusions

In two decades of development, ML/MIR have been growing exponentially as research fields, not only in the academic sphere but also in the industrial/commercial scenario. However, despite research in these fields is in constant expansion, transforming the possibilities placed to musical practices to boost research with very diversified creative biases, there is still a lack of works concerned with carrying out reflections on the compositional exploration of these tools.

We aim to contribute with understanding the creative exploration of these tools in contemporary creative practices, outlining and scrutinizing how composers have been applying these resources in musical creation in the

context of the composition of pieces that combine instrumental/vocal and sound processing in real-time. By identifying the previously listed paradigms, it was possible to glimpse not only a series of technical frameworks but also to point out compositional approaches used by composers as strategies to solving technical-musical issues that aim to achieve certain poetic conceptions and aesthetic results.

As already highlighted, the generalization of these application strategies does not aim to exhaust or to rigidly categorize all possibilities of explorations of MIR/ML methods in the context of live-electronics/interactive music composition. As stated, the objective is to circumscribe in a general way, ways of thinking and conceiving the creative use of ML/MIR, seeking to offer a broader view of the use of these resources that can prove to be useful in analytical, creative, and pedagogical contexts.

References

- [1] Nick Collins. Automatic Composition of Electroacoustic Art Music Utilizing Machine Listening. *Computer Music Journal*, 36(3):8–23, September 2012.
- [2] Nick Collins. Machine Listening in SuperCollider. In *The SuperCollider book*. MIT Press, Cambridge, Mass, 2011.
- [3] Albert S. Bregman. *Auditory Scene Analysis*. The MIT Press, 1990. Publisher: The MIT Press.
- [4] István Winkler, Susan Denham, Robert Mill, Tamás M. Bóhm, and Alexandra Bendixen. Multistability in auditory stream segregation: a predictive coding view. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1591):1001–1012, April 2012.
- [5] Alexander Gutschalk and Andrew R. Dykstra. Functional imaging of auditory scene analysis. *Hearing Research*, 307:98–110, 2014. Place: Netherlands Publisher: Elsevier Science.
- [6] Robert Rowe. *Interactive music systems: machine listening and composing*. MIT press, Cambridge, 1993.

- [7] Joe Futrelle and J. Downie. Interdisciplinary Research Issues in Music Information Retrieval: ISMIR 2000. *Journal of New Music Research*, 32:121–131, June 2003.
- [8] Eduardo Reck Miranda and Marcelo Wanderley. *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*. A-R Editions, Inc., Middleton, Wis, 1st edition edition, July 2006.
- [9] Song Wang, Marcelo Wanderley, and Gary Scavone. The Study of Mapping Strategies Between the Excitators of the Single-Reed Woodwind and the Bowed String. In *Proceedings of the 7th Conference on Sound and Music Technology (CSMT)*, pages 107–119. Springer, January 2020.
- [10] Adriano Claro Monteiro. *Criação e performance musical no contexto dos instrumentos musicais digitais*. PhD thesis, 2012. Accepted: 2018-08-20T07:47:37Z Publisher: [s.n.].
- [11] Gabriel Rimoldi. *Metaflauta : uma abordagem ecológica aplicada ao design e processos criativos a partir de um instrumento expandido*. PhD thesis, 2018. Accepted: 2018-09-04T18:43:32Z Publisher: [s.n.].
- [12] Andy Hunt, Yo Dd, Marcelo Wanderley, and Matthew Paradis. The Importance of Parameter Mapping in Electronic Instrument Design. *Journal of New Music Research*, 32, May 2002.
- [13] Fernando Iazzetta. Meaning in Musical Gesture. In *Trends in Gestural Control of Music [CD-ROM]*, pages 259–268. IRCAM, January 2000.
- [14] Hélena Regius and Joakim Langer. *Kung Kalle av Kurrekurreduttön*. Fórum Bokförlaget, 2002.
- [15] Anna Einarsson. *Singing the body electric - Understanding the role of embodiment in performing and composing interactive music*. PhD thesis, Lund University Royal College of Music Stockholm, October 2017.
- [16] Anders Friberg, Erwin Schoonderwaldt, and Patrik Juslin. CUEx: An Algorithm for Automatic Extraction of Expressive Tone Parameters in Music Performance from Acoustic Signals. *Acta Acustica united with Acustica*, 93:411–420, May 2007.
- [17] Anna Einarsson and Anders Friberg. Using Singing Voice Vibrato as a Control Parameter in a Chamber Opera. In *41st International Computer Music Conference*, pages 18–21, University of North Texas, September 2015.
- [18] 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, April 2002.
- [19] Mark Johnson. *The Body in the Mind: The Bodily Basis of Meaning, Imagination, and Reason*. University of Chicago Press, 1ª edição edition, January 2013.
- [20] Mikhail Malt and Marta Gentilucci. Real Time Vowel Tremolo Detection Using Low Level Audio Descriptors. *arXiv:1511.07008 [cs]*, November 2015. arXiv: 1511.07008.
- [21] Miller Puckette, Theodore Apel, and David Zicarelli. Real-time audio analysis tools for Pd and MSP. In *Proceedings of International Computer Music Conference*, São Francisco, October 1998.
- [22] Arshia Cont. Antescofo: Anticipatory Synchronization and control of Interactive parameters in Computer Music. In *Proceedings of International Computer Music Conference*, Belfast, 2008.
- [23] Philippe Manoury. La note et le son, 1990.
- [24] Andrew May. Philippe’s Manoury Jupiter. In *Analytical Methods of Electroacoustic Music*, pages 145–185. Routledge, December 2005.
- [25] Mikhail Malt. Khorwa: A musical experience with ‘autonomous agents’. In *ICMC*, 2004.
- [26] José Miguel Fernandez, Jean-Louis Giavitto, and Pierre Donat-Bouillud. AntesCollider: Control and Signal Processing in the Same Score. June 2019.
- [27] José H. P. Velloso. Representação, intuição e contato na composição com algoritmos. Master’s thesis, Universidade Estadual de Campinas, Campinas, 2009.
- [28] Joel Chadabe. Interactive Composing: An Overview. *Computer Music Journal*, 8(1):22–27, 1984. Publisher: The MIT Press.
- [29] Philippe Manoury. Sur la synthèse 3F, 2020.

Acalma-te: resultados de composição automática e iterativa para relaxamento

Cláudio Gomes¹

¹Departamento de Ciências Exatas e Tecnologias - Universidade Federal do Amapá (UNIFAP)
Caixa Postal 261 – 68.906-970 – Macapá – AP – Brazil

claudiorogério@unifap.br

Resumo. A música é uma forma de expressão invisível, não palpável e ainda maleável tornando-se difícil a tarefa de descrevê-la, rotulá-la ou construí-la. A base de qualquer música é composta por o ritmo, melodia e harmonia. Este trabalho apresenta uma composição intitulada de Acalma-te, resultado do projeto experimental de geração de músicas para relaxamento. O compositor informa, tonalidades, bpm, quantidade e tipo de estrofes e quantidade de faixas. Pode-se acompanhar o resultado em tempo real permitindo interatividade para mudanças de tonalidades, bpm e qualquer outra simbologia necessária para o compositor. Ao final, é gerado um arquivo MIDI de cada faixa como de todas as faixas podendo adicioná-las em qualquer programa de edição sonora.

Abstract. Music is an invisible, intangible form of expression, making it difficult to describe, label or construct it. Any song is composed of rhythm, melody and harmony. This work presents a composition entitled Acalma-te (Calm down), a result of the experimental project of generating music for relaxation. The composer informs, key, bpm, number and type of verse and number of tracks. It should follow the result in real time allowing interactivity to change keys, bpm and any other symbology needed by the composer. At the end, a MIDI file is generated for each track as for all tracks, which can be added in any sound editing program.

1 Introdução

As ferramentas atuais de composição musical apresentam diversos desafios para criatividade em determinados propósitos, embora sejam eficientes para o processo de composição musical. Dentre várias problemáticas, há duas principais discussões estão voltadas para o significado e pensamento musical, que ao passarem pelo desenvolvimento de ferramentas tecnológicas, não apresentam resultados que acompanhem direções inicialmente idealizadas pelo compositor [1, 2, 3, 4].

O trabalho de Raglio *et. al.* [5] relata que os humanos não conseguiram distinguir a diferença de composições algorítmicas e humanas. Para isso, realizou quatro experimentos: ouvintes que preferiam músicas ou algoritmos musicais, e em outro grupo os ouvintes escolhiam se preferem composições criadas e performadas por humano, composições por humano e performadas pela máquina, composições criadas por máquinas e performadas pelos humanos e composições criadas e performadas por máquinas. Tiveram a avaliação de o total de 500 voluntários.

Já o trabalho de Williams *et. al.* [6] descreve um estudo de caso dirigido para ansiedade e estresse através de composições geradas por inteligência artificial. Utilizou como entradas para seu algoritmo as características psicológicas emocionais dos participantes através de um questionário identificando correlações inversas ao espaço afetivo cartesiano.

Este trabalho apresenta resultados do projeto intitulado Acalma-te, uma proposta inicial de ferramenta de composição facilitada pelo computador. O compositor define diversas características como: faixas, BPM, estrofes, campo harmônico e que permite a adaptação no decorrer da evolução da sonoridade. Nesse momento, esse projeto destinou-se na criação de músicas para relaxamento para auxiliar a sociedade durante períodos de isolamento social. Ao final da composição, é gerado um arquivo MIDI com todas as faixas permitindo a posterior edição, reconstrução e adaptação. O artigo está estruturado da seguinte forma: a seção 2 apresenta-se a estrutura do projeto Acalma-te. Na seção 3, apresenta-se as conclusões.

2 Acalma-te

O projeto Acalma-te parte do princípio que o processo de relaxamento é o momento em que a pessoa esquece de seus problemas, medos, incertezas, dúvidas ou qualquer que seja o seu motivo, para atentar-se e focar-se na melodia e harmonia musical. No entanto, a pessoa não ficará à eternidade nesse período e logo voltará a lembrar-se de seus questionamentos iniciais. É nesse momento que a música modifica alguma características importante como a mudança de tonalidade ou escala musical a fim de manter a atenção novamente do ouvinte. Ou seja, o processo de relaxamento é um curto período de tempo em que permitiu-se isolar-se de problemas sociais ou pessoais. Neorologicamente, o cérebro atua como um receptor de todos os sentidos, nesse caso a audição, e ao escutar uma música tenta decifrar suas características. Para algumas pessoas, essa tarefa pode durar segundos, outras vários minutos e, quando o cérebro aprende ou desiste de determinado padrão musical, é o momento de alterar as características musicais, forçando novamente a curiosidade do cérebro.

O projeto Acalma-te constitui de um modelo de composição algorítmica direcionado para músicas relaxantes. Nesse momento, não trabalhou-se com características rítmicas. Utilizou-se a linguagem ChuckK e Python. A linguagem ChuckK foi responsável por toda a codificação de composição e Python para a geração de arquivos MIDI. Sobre a codificação de composição, criou-se

variáveis que são definidas como *Notas, Escalas e Frases*. Conforme o algoritmo 1, criou-se duas trilhas a partir da variável *Nota*, que estão conectados ao conversor digital analógico (DAC). Com as variáveis criadas, pode-se criar notas em modelo totalmente flexível para qualquer tipo de característica fundamental seja por nota ou acorde, duração da nota, volume, bpm, frequência base e tipo de timbre. Na linha 7, criou-se o tipo de escala musical e tonalidade. Até o momento, implementou-se os modelos de escala gregorianos.

Seja para alegria, tristeza, dúvidas, medos cada escala gregoriana tem um objetivo principal e maleável de intenção musical. Ou seja, a escolha da escala musical indica a intenção formal da música gerada pelo compositor. Além disso, para evitar a possível monotonia ou correspondência pode-se alterar ações em tempo real, tanto pelo compositor ou pelo algoritmo.

É possível extrair as sequências dos acordes, pentatônicas, sétimas e aleatórias dentro de cada campo escolhido. Na linha 8, criou-se duas estrofes na qual há frases pré-definidas como *graves e agudas*. O compositor pode editar e criar a quantidade de frases que desejar, ou ainda deixar para que seja criada conforme as características de campo harmônico, timbre e bpm.

No exemplo ilustrado no algoritmo 2, uma única estrutura de execução de uma pequena peça de cada faixa tocadas sequencialmente. Após tocar as notas correspondentes de cada faixa, atualiza-se informações das escalas (linhas 4–5), frases (linhas 6–7) e notas (linhas 8–9) com dados atualizados das escalas e frases ou ainda atualizar a nota para acorde.

Além disso, o ChucK permite funcionalidades para que se toque faixas em paralelo que não foram apresentadas no artigo, porém são todas aplicadas nos resultados¹. Ou seja, pode-se criar várias estruturas como no exemplo do algoritmo 2 responsáveis para cada faixa musical com suas notas, escalas e frases pré-definidas. Para cada faixa criada, emula-se como um instrumento MIDI permitindo a edição em qualquer ferramenta que possua entradas instrumentos MIDI.

Ao final, gera-se um arquivo MIDI com todas as faixas criadas pelo compositor para possíveis ajustes rigorosos de edição. O ChucK não apresenta a funcionalidade de arquivo MIDI, portanto, realizou-se a intercomunicação entre ChucK e Python a partir de um arquivo intermediário, servindo de entrada para a biblioteca Midiutil [7] gera-se o resultado em arquivo MIDI.

3 Conclusão

O projeto Acalma-te tem o objetivo de geração musical para relaxamento como auxílio nos momentos de isolamento social. Para trabalhos futuros pretende-se realizar testes iniciais em clínicas, hospitais ou centros de tratamentos especializados ao tópico e por fim, criar uma plataforma de áudio streaming pública e gratuita adaptativa ao

contexto dos ouvintes.

Algorithm 1 Exemplo 1

```

1: Nota track1, track2;
2: track1.connect( dac );
3: track2.connect( dac );
4: track1.set_nota( 'D5', 'colcheia', 0.5, bpm, '432Hz',
   'flauta' );
5: track2.set_nota( 'D4m', 'colcheia', 0.5, bpm, '432Hz',
   'flauta' );
6: Scale seq1;
7: seq1.set_scale( 'frigio', 'Bb' );
8: Phrases tempos[2];
9: tempos[0].set_phrase( 'graves' );
10: tempos[1].set_phrase( 'medio' );

```

Algorithm 2 Exemplo 2

```

1: for i = 1 to N do
2:   track1.play();
3:   track2.play();
4:   seq1.next();
5:   seq2.next();
6:   tempos[0].next_tempo();
7:   tempos[1].next_tempo();
8:   track1.set_nota( seq1.get_note(),      tem-
     pos[0].get_phrase,
     0.2, bpm, '432Hz', 'flauta' );
9:   track2.set_nota(seq2.get_note()+seq2.get_chord(),
     tempos[1].get_phrase, 0.2, bpm, '432Hz', 'flauta' );
10: end for

```

Referências

- [1] Rodolfo Coelho de Souza and Regis Rossi Alves Faria. Oito reflexões sobre a criatividade na composição auxiliada por computadores. 2011.
- [2] Omar Lopez-Rincon, Oleg Starostenko, and Gerardo Ayala-San Martín. Algorithmic music composition based on artificial intelligence: A survey. In *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, pages 187–193. IEEE, 2018.
- [3] Iyad Abu Doush and Ayah Sawalha. Automatic music composition using genetic algorithm and artificial neural networks. *Malaysian Journal of Computer Science*, 33(1):35–51, 2020.
- [4] Roger T Dean. *The Oxford handbook of algorithmic music*. Oxford University Press, 2018.
- [5] Alfredo Raglio, Paola Baiardi, Giuseppe Vizzari, Marcello Imbriani, Mauro Castelli, Sara Manzoni, Francisco Vico, and Luca Manzoni. Algorithmic music for therapy: effectiveness and perspectives. *Applied Sciences*, 11(19):8833, 2021.
- [6] Duncan Williams, Victoria J Hodge, and Chia-Yu Wu. On the use of ai for generation of functional music to improve mental health. *Frontiers in Artificial Intelligence*, 3:94, 2020.
- [7] MIDIUtil. <https://media.readthedocs.org/pdf/midiutil/1.0.1/midiutil.pdf>. Acesso em 22.09.2021.

¹<https://github.com/claudiorogero/acalma/>

Short Papers

Investigando Tecnologias Educacionais para o Ensino e Aprendizagem de Música para Deficientes Visuais Totais

Sara Elene Lobato dos Remedios¹, Sandro Luís Freire de Castro Silva¹, Simone Bacellar Leal Ferreira¹

¹Programa de Pós-Graduação em Informática - Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Av. Pasteur, 458, 22290-255-Rio de Janeiro, RJ

sara.lobato@edu.unirio.br, sandro@edu.unirio.br, simone@uniriotech.br

Abstract. *O surgimento de tecnologias assistivas para o ensino de música para cegos trouxe inúmeros benefícios, porém, mesmo com a evolução tecnológica, os diferentes métodos de ensino podem não abranger a totalidade desse grupo. Portanto, se faz necessário conhecer as tecnologias disponíveis, para os pontos de convergência sejam conhecidos e novas funcionalidades sejam propostas. Para essa tarefa, este trabalho tem como objetivo investigar na literatura as tecnologias educacionais que apoiam o ensino e aprendizagem de música para deficientes visuais totais por meio de um mapeamento sistemático da literatura. Este trabalho apresentou um levantamento de tecnologias voltadas para músicos e alunos de música deficientes visuais.*

1. Introdução

Segundo o último censo realizado no Brasil 3,5% da população possui algum tipo de deficiência visual, e desse grupo, cerca de 528.624 pessoas são incapazes de enxergar (cegos) [1]. Assim como outros grupos de pessoas com deficiências, os cegos no Brasil têm de conviver diariamente com a dificuldade da garantia de direitos, especialmente em temas como mobilidade urbana, saúde e educação.

Apesar dos desafios, nos últimos anos, iniciativas significativas foram tomadas para o acesso à educação de pessoas cegas. Um exemplo é a criação da Portaria do Ministério da Educação (MEC) n. 3.284, de 07 de novembro de 2003, que determina requisitos de acessibilidade e garante que alunos deficientes visuais tenham acesso a todos tipos de recursos, seja tecnologias assistivas ou adaptações pedagógicas, que garantam sua educação básica [2]. Pode-se dizer que por meio de políticas públicas, muitos deficientes visuais garantiram seu direito à educação, porém, acredita-se que um fator pode ter sido decisivo para apoiar esse avanço: o desenvolvimento tecnológico.

Em meio às diversas formas de promover a inclusão à educação por meio de tecnologias assistivas, destaca-se a presença do ensino de música como um forte aliado. Nesse caso, duas ferramentas, tecnologia e música, unem-se no processo ensino-aprendizagem de um público específico: alunos de música e deficientes visuais. É comum que essas pessoas aprendam algum instrumento musical e toquem com excelência pois são as que mais possuem o “ouvido absoluto” comparadas às pessoas videntes [3, 4]. Antes das tecnologias com finalidade de auxiliar o ensino-aprendizagem de música, a audição de uma pessoa nessas condições era o único recurso que ela dispunha para aprender música, o que era um grande limitador para a inserção dessas pessoas no ambiente de aprendizagem.

Por meio de tecnologias educacionais que apoiam o ensino e aprendizagem, é possível que esse público específico não só aprenda melodias, mas também crie suas partituras, permitindo que o aluno com limitações visuais possa exercer sua capacidade criativa. Outrossim, esse tipo de tecnologia possibilita que tanto o aluno de música deficiente visual quanto o aluno vidente executem as mesmas atividades durante a aula de música simultaneamente, assegurando na prática a acessibilidade [5].

A profusão de tecnologias assistivas para o ensino de música trouxe inúmeros benefícios, porém, mesmo com a evolução tecnológica, os diferentes métodos de ensino e abordagens podem não abranger totalidade desse grupo, o que torna necessário conhecer essa gama de tecnologias disponíveis, para que a partir disso, os pontos de convergência sejam conhecidos e novas funcionalidades sejam propostas. Para essa tarefa, este trabalho tem como objetivo investigar na literatura as tecnologias educacionais que apoiam o ensino e aprendizagem de música para deficientes visuais totais por meio de um mapeamento sistemático da literatura.

Além desta seção de introdução, este artigo está organizado da seguinte forma: a Seção 2 traz a metodologia do estudo; na Seção 3, é apresentado o resultado do mapeamento; a Seção 4 traz a discussão dos trabalhos encontrados; e, por fim, a Seção 5 conclui o artigo.

2. Metodologia

A pesquisa foi conduzida com base em diretrizes para realização de mapeamentos sistemáticos da literatura (MSL) [6, 7]. As diretrizes propõem a execução das seguintes etapas: i) planejamento da pesquisa; ii) definição das questões de pesquisa; iii) elaboração da string de busca; iv) critérios de inclusão e exclusão; v) critérios de seleção; e vi) extração de dados.

A questão de pesquisa (QP) que apresenta os objetivos deste estudo foi formulada de acordo com os critérios especificados no PO (*Population, Outcomes*). Onde: **P** - Estudos que abordam ensino de música para cegos e; **O** - Artefatos tecnológicos voltados para o ensino de música para cegos.

O objetivo do estudo foi definido com base no GQM (*Goal-Question-Metric*) [8]: **analisar** estudos que abordam o ensino de música para deficientes visuais totais **com o propósito de** conhecer as tecnologias utilizadas no ensino de música **para** esse público específico (cegos) **do ponto de vista de** pesquisadores no contexto de sistemas

de informação. As buscas foram feitas nas bibliotecas digitais ACM e Web of Science.

2.1. Definições das Questões de Pesquisa

A questão de pesquisa (QP) deste estudo foi: “*Quais os softwares mais utilizados para o ensino de música para cegos?*”. Para complementar a questão de pesquisa, foram definidas subquestões (Sub-Q) para que as informações coletadas fossem analisadas a partir destas questões: “*Quais as principais características identificadas nos softwares?(Sub-Q1)* e “*Quais as principais dificuldades relatadas pelos autores?(Sub-Q2)*”.

2.2. Elaboração da String de Busca

A string definida foi: ((“software” or “technology”) AND (“music”) AND (“blind”) AND (“education”)).

2.3. Critérios de Inclusão e Exclusão

Para selecionar os artigos retornados pela string de busca nas bibliotecas digitais, foram definidos critérios de inclusão (CI) e critérios de exclusão (CE). Para ser aceito, o artigo bastava atender ao menos um dos CI e para ser excluído bastava atender ao menos um dos CE. Os critérios estão descritos na Tabela 1.

Tabela 1: Critérios de Inclusão e Exclusão

ID	Critério de Inclusão
CI1	O estudo apresenta alguma tecnologia voltada para o ensino de música para deficientes visuais.
CI2	O estudo avalia uma tecnologia de ensino de música voltada para o público de deficientes visuais.
CI3	O estudo deixa claro quais os artefatos tecnológicos utilizados para a execução da pesquisa.
#	Critério de Exclusão
CE1	O artigo não menciona tecnologias de música para deficientes visuais.
CE2	O estudo não é um estudo primário.
CE2	O estudo é antigo e descontextualizado.

2.4. Critérios Seleção

Os estudos foram selecionados por meio das seguintes etapas: (1) busca nas bibliotecas digitais; (2) a leitura de título, resumo e palavras-chave; (3) leitura da introdução e conclusão; (4) leitura completa; e (5) extração dos dados. Para a execução da etapa 4 foram definidos critérios de qualidade (CQ) para analisar se os resultados obtidos respondem às questões necessárias para esta pesquisa. Os estudos que não corresponderam a todos os CQ definidos, foram excluídos. Os critérios de seleção são apresentados na Tabela 2.

Tabela 2: Critérios de Qualidade

CQ	Descrição
CQ1	O estudo define claramente o objetivo da pesquisa (define questão de pesquisa)?
CQ2	O estudo apresenta questões relevantes quanto ao ensino de música para cegos.
CQ3	O estudo deixa claro quais os artefatos tecnológicos utilizados para a execução da pesquisa.

2.5. Extração dos Dados

Os dados foram extraídos e registrados em uma planilha eletrônica com: a) um código de identificação para o estudo (ID); b) o título do estudo; c) o ano em que foi publicado; d) os autores do estudo; e) onde o estudo foi publicado; f) quais os stringsoftwares o estudo menciona; e e) quais as dificuldades relatadas.

Apesar deste trabalho ter seguido minuciosamente todas as etapas da metodologia para garantir a confiabilidade, podem existir fatores que tornem os resultados incertos. Para isso, os CQ foram empregados pelos pesquisadores buscando reduzir erros nos textos dos trabalhos analisados que possam ter acarretado inconsistências nas análises de resultados. Outro fator a ser levado em consideração é a presença de apenas dois pesquisadores para validar os resultados obtidos.

3. Resultados

Primeiramente foi executada a string de busca nas bases; o total de trabalhos retornados foi de 548, sendo retirados os duplicados, restando 533. Feito isso, aplicou-se o primeiro filtro através dos CI e CE e da leitura de título, resumo e palavras-chave, restando apenas 25 estudos. O segundo filtro foi aplicado por meio da leitura da introdução e conclusão, restando 19 estudos. O terceiro filtro então foi aplicado fazendo a leitura completa dos estudos e seguindo os CQ, finalizando então com 13 estudos. Após essa etapa, foi iniciada a extração dos dados.

Os estudos selecionados foram publicizados¹. Os dados extraídos buscaram responder à QP e SubQ's. Os seguintes softwares utilizados para o ensino e aprendizagem de música, para cegos, foram identificados: Braille Music Editor (com versão gratuita e paga), Toccata (comercial), Goodfeel (comercial), RS Braille, MusiBraille (gratuito), InvisibleSound, Musica Parlata e Zero-Vision. A Tabela 1 apresenta os softwares encontrados, e os estudos de referência.

Tabela 3: Softwares mais utilizados para o ensino de música para cegos (QP)

Software	Estudo
Braille Music Editor	E11 e E12
Toccata	E11 e E12
Goodfeel	E1, E11 e E12
RS Braille	E13
MusiBraille	E9 e E11
InvisibleSound	E4
Musica Parlata	E10
Zero-Vision	E12

Os estudos E11 e E12 apresentam diversos softwares para ensino de música, destacando o Braille Music Editor, Toccata e o Goodfeel como programas conhecidos para transcrever músicas em Braille. O E12 sugere

¹https://docs.google.com/spreadsheets/d/e/2PACX-1vQrbkLqBfewgYXHWauJwn4sXkg9FBaj6_R7skiSgZJcxBIhGhhXr6888cHCWRBSg2divrQS-dfPXkL1/pubhtml

o *software* Zero-Vision, como uma alternativa, pelo fato de possuir um sequenciador MIDI, que permite ao músico cego fazer suas próprias produções musicais. O mesmo estudo menciona os *softwares* Toccata e Goodfeel, pois são usadas instâncias presentes nestes *softwares* para a criação do Zero-Vision. O E11 apresenta de forma detalhada o projeto Musibaille, um *software* voltado para o ensino e aprendizagem de música para o público deficiente visual, sendo um dos programas mais utilizados no Brasil.

No que diz respeito às contribuições metodológicas atreladas a tecnologia para o ensino de música para cegos, o E1 apresenta uma pesquisa com 11 profissionais, produtores e compositores, cegos, para compreender como realizam seu trabalho e identificar tecnologias que atendam às suas necessidades. Outro trabalho relevante é E10, que propõe métodos para deficientes visuais estudarem música sem usar o Braille, propondo a metodologia Musica Parlata, que faz com que o usuário cego cante o nome das notas, trazendo a percepção direta que se tem com uma partitura.

Ainda no que diz respeito aos aspectos metodológicos do ensino, no estudo E3 os autores apresentam uma tecnologia para ensinar harmonização melódica e tonal, para recomposição musical e escuta de partituras, tendo como uma das finalidades ajudar crianças cegas a andar em linha reta. E em E9 é proposta uma metodologia de ensino de música, baseada no uso do *software* Musibaille, para crianças cegas.

Um ponto de destaque é que alguns trabalhos apresentam questões relacionadas à profissionalização desses músicos, como por exemplo E7 onde são discutidas descobertas sobre escolaridade, experiências de vida de músicos amadores, professores de música cegos e videntes que trabalham com alunos com deficientes visuais. Além disso o trabalho E5 explica o processo de digitalização de partituras em Braille para a seção de música de uma biblioteca nacional para cegos, o que pode ser útil para esses alunos quando se tornarem profissionais. Alguns dos artigos não apresentam um *software* específico, no entanto destacam requisitos importantes que um *software* desse tipo deve dispor para usuários deficientes visuais.

3.1. Quais as principais características identificadas nos softwares? (Sub-Q1)

Em meio aos principais aspectos que os *softwares* possuem, o sintetizador de voz, por meio de um leitor de tela, é um dos mais importantes. O trabalho E1 aponta as melhorias na acessibilidades desses programas, e relata que muitas empresas comerciais têm investido em tecnologias de leitores de tela, fazendo com que essa iniciativa seja considerada uma tendência. As pesquisas E9 e E11 também abordam a funcionalidade de *feedback* de áudio e leitor de tela em todas as funções.

O trabalho E2 apresenta uma série de requisitos que um *software* voltado para músicos cegos, e enfatiza o sintetizador de voz, através do acesso a tela (ex: NVDA ou JAWS). Outro requisito apontado é a inserção de notas e outros caracteres, inserção de comandos pelo teclado, e

criação de atalhos no teclado para a execução dessas ações, porém, essas teclas de atalho não devem coincidir com as padrão do sistema operacional do computador. Outros requisitos que o trabalho E2 aponta é a possibilidade de importar e exportar textos de música para arquivos de outras extensões, além da possibilidade de converter partituras musicais digitalizadas para formatos acessíveis, presentes nos *softwares* mencionados nos trabalhos E9 e E11.

Os trabalhos E10, E11 e E13 mencionam a presença de um menu com funções típicas de outros aplicativos como acesso às opções de configuração e ferramentas de ajuda. O *software* mencionado na pesquisa E10 contém materiais de áudio especial com as notas solfejadas, ou seja, “cantadas” no seu tom correto e é possível também adicionar outra informação, por exemplo os acordes ou o clique do metrônomo.

3.2. Quais as principais dificuldades relatadas pelos autores? (Sub-Q2)

Os autores dos estudos destacam dificuldades em ambientes de ensino de música e *softwares* mencionando-os em seus trabalhos. O trabalho E1 mostra que profissionais que produzem música precisam de programas específicos para executarem seu trabalho, mas se esse profissional for cego, ele possivelmente encontrará dificuldades na hora da produção das músicas, pois a maioria dos programas utilizados para essa atividade não é acessível, o que os leva a buscar um *software* adicional, na maioria das vezes com leitor de tela, para conseguir realizar suas tarefas.

O trabalho E4 reflete alguns argumentos para justificar o mercado pequeno de tecnologias acessíveis, como o valor da implementação, muitas vezes sem retorno financeiro. Outro argumento utilizado é que não há público suficiente que faça com que o desenvolvimento desse tipo de tecnologia seja rentável para as empresas fornecedoras. No entanto, os autores refutam esses argumentos, já que os custos de investimento em recursos de acessibilidade diminuíram no decorrer dos anos, tornando viável o investimento em tecnologias acessíveis.

A pesquisa E2 relata dificuldades em ambientes de ensino de música no momento da troca de informação entre o professor vidente e o aluno cego, pois o aluno cego requer um professor com habilidades específicas. A pesquisa E11 também menciona essa dificuldade e destaca que professores de música às vezes se recusam a lecionar para alunos cegos, por não se sentirem aptos para tal atividade. O mesmo pode ocorrer quando o professor cego precisa lecionar para o aluno vidente, é provável que o professor tenha dificuldades na hora de explicar símbolos musicais. Assim, os autores do E2 destacam que o *software* usado em ambientes de ensino de música deve ser útil para ambos os grupos.

O trabalho E10 aponta duas limitações do *software* Musica Parlata. A primeira é a dificuldade de “ler” em paralelo vários versos melódicos. A outra limitação é que, apesar dos usuários cegos que testaram o *software* considerarem o programa fácil de usar, houveram alguns

relatos de dificuldades, como a presença de ecos que dificultaram a leitura das partituras.

4. Discussão

Os trabalhos analisados possibilitaram reflexões importantes no que diz respeito à acessibilidade em tecnologias musicais. O E1 trata essa questão como elemento central, e mostra preocupação com a alocação profissional do egresso cego de um curso de música mediado por tecnologia. A questão é que mesmo após aprenderem música e se alocarem profissionalmente, as dificuldades permanecem, fazendo com que estes profissionais tenham de recorrer a diversas ferramentas tecnológicas para produção musical, sinalizando que esses *softwares* não possuem nenhuma integração.

Pode-se dizer que a maioria desses *softwares* analisados precisam de melhorias em acessibilidade. Esse fenômeno faz com que os músicos cegos busquem *softwares* adicionais que tenham o recurso de leitor de tela e um suporte mais inclusivo. Por meio da análise, pode-se também observar que os músicos cegos precisam recorrer à ajuda de colegas videntes, pela complexidade de alguns *softwares*. O trabalho E1 abre portas para serem discutidas inovações nos programas de produção musical e composição. O intuito portanto, não é observar esses *softwares* exclusivamente para cegos, mas que os existentes sejam inclusivos, e possam ser usados por seus colegas com visão, já que eles realizam a mesma atividade.

No E4 os pesquisadores comentam sobre a escassez de *softwares* acessíveis e apontam que pouco apoio financeiro para implementação ou público alvo escasso são os principais argumentos utilizados para justificar essa carência. Contudo, esses argumentos podem ser considerados ultrapassados, uma vez que cada vez os custos para a produção de softwares têm diminuído, e também porque pessoas com deficiência estão presentes no mercado de trabalho e são produtivas.

O E10 apresenta o *software* e metodologia de ensino de música para deficientes visuais MUSICA PARLATA, que oferece ao aluno deficiente visual uma percepção auditiva das notas ao solfeja-las em suas respectivas linhas melódicas e reproduz o nome dos acordes. Esse método se mostra eficiente para decorar as notas musicais, aprender melodias, porém no que diz respeito à inclusão deixa a desejar, pois ele ignora a aprendizagem da musicografia Braille (teoria musical), e foca apenas no aprendizado prático, o que não acontece em salas de aula de música, onde as aulas são o conjunto de atividades teóricas e práticas.

Outro ponto relevante a ser discutido trazido pelo E5, é a formação de professores de música. Nem todo professor, que leciona em cursos de música, possui graduação em Música. Existem instituições de ensino de música onde não é necessário que professores de música tenham nível superior, apenas o curso de nível técnico é suficiente para ministrar aulas, porém em escolas regulares é indispensável o diploma de graduação. No entanto, os cursos

de graduação de música, nem sempre disponibilizam ao menos uma disciplina voltada para prepará-los para atender alunos com limitações.

O E11 relata que essa falta de preparo é motivo de professores de música se recusarem a lecionar para alunos com necessidades especiais, por não se sentirem capacitados, por isso um dos objetivos do projeto apresentado no trabalho, o MusiBraille é proporcionar um treinamento para educadores de modo a introduzi-los ao *software* e prepará-los para usar em sala de aula.

5. Considerações Finais

Este trabalho apresentou um levantamento de tecnologias voltadas para músicos e alunos de música deficientes visuais. Pode-se constatar que são poucos os *softwares* comercializados ou acessíveis, e que muitos possuem limitações a serem resolvidas, além de outros ainda estarem em fase de testes.

Apesar da diversidade de ferramentas direcionadas a músicos e alunos de música cegos, e elas possuírem diversos pontos em comum, é primordial o aprimoramento de pesquisas que os sustentem e aperfeiçoem, dada a complexidade para se construir *softwares* como esses. Portanto, as evoluções dessas tecnologias devem ser contínuas, uma vez que podem ser listadas várias oportunidades de melhoria e aperfeiçoamento.

Como trabalhos futuros pretende-se fazer uma avaliação da interação de alunos de música, deficientes visuais totais, especificamente com os *softwares* para analisar as expectativas e necessidades desses usuários, identificando as principais barreiras enfrentadas por eles e a partir disso propor um ambiente de ensino mais adaptado.

Referências

- [1] Instituto Brasileiro de Geografia e Estatística (IBGE). Censo Brasileiro de 2010. Technical report, Instituto Brasileiro de Geografia e Estatística, Rio de Janeiro, 2012.
- [2] Brasil. Portaria nº 3.284, de 7 de novembro de 2003. *Diário Oficial da República Federativa do Brasil*, 2003.
- [3] Roy H Hamilton, Alvaro Pascual-Leone, and Gottfried Schlaug. Absolute pitch in blind musicians. *Neuroreport*, 15(5):803–806, apr 2004.
- [4] Emílio Figueira. *A presença da Pessoa com Deficiência Visual nas Artes - Projeto Ebooks*. 2009.
- [5] Artur Batista de Oliveira Rocha. O Ensino da Música para Deficientes Visuais. Revista Científica Multidisciplinar Núcleo do Conhecimento. *Revista Científica Multidisciplinar Núcleo do Conhecimento*, 5(8):105–120, Novembro 2017.
- [6] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering. Technical Report EBSE 2007-001, Keele University and Durham University Joint Report, 2007.
- [7] Kai Petersen, Sairam Vakkalanka, and Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, 64:1–18, 2015.
- [8] Victor R. Basili. Software modeling and measurement: The goal/question/metric paradigm. Technical report, USA, 1992.

Production of digital content in music teacher education: a study about podcast's possibilities

Marcos da Rosa Garcia¹, Gutenberg Lima Marques², Matheus Barros³, Juciane Araldi Beltrame²

¹ Instituto Federal da Paraíba
Av. Primeiro de Maio, 720, 58015-435 - João Pessoa - PB

² Laboratório de Tecnologias e Educação Musical (Tedum/UFPB) - Departamento de Educação Musical da
Universidade Federal da Paraíba
Cidade Universitária, 58051-900 - João Pessoa - PB

³ Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano
R. Maria Luzia de Araújo Gomes Cabral, 791, 56316-686 - Petrolina - PE

{marcos-rosa, gutenberglm, matheus.barros, juciane.beltrame}@emo.ufpb.br

Abstract. This short paper presents an ongoing research that intertwines the theme of educational digital content production for the internet, specifically the audio podcast format, with the pedagogical practices developed in the context of music teacher education and emergency remote teaching. We aim at analyzing the experience of producing digital pedagogical-musical content in the podcast format by students of two Music Education Degree courses. The study uses a qualitative approach and the methodological strategy is based on concepts of action-research. The research is being developed by Technologies and Music Education Research Group (Tedum-UFPB) and by a team of professors from two federal higher education institutions. Data collection will be carried out through the development of field diaries by the research team and through conversation roundtables with the participant students, besides the entire process of documentation, registration and analysis of the phases that make up the action-research cycle. The research presented here can contribute to the processes of creation and conception of audio format content, seeking methodologies that are specific to the musical field, enhancing collective spaces for creation, valuing different authorships and encouraging pedagogical and musical diversity.

1. Background information

The health crisis generated by the Covid-19 pandemic had instant effects in the field of education. Isolation and social distancing measures forced the abrupt suspension of classes and the closing of schools and universities. The appropriation of digital media for the continuation of classes and the teaching of content was presented as an alternative most welcomed by educational systems in order to overcome the imposed situation [1].

Thus, this ongoing research* interweaves the theme of educational digital content production for the internet, specifically the audio podcast format, with the pedagogical practices developed in the context of music teacher education and emergency remote teaching.

A podcast "is a digital media file, or a collection

of related files, [...] for playback on portable media players and personal computers" [2]. The production process of a podcast can mobilize a diversity of knowledge: about the subject that will be dealt with, as well as the characteristics of this format, in addition to the technical knowledge of recording. According to Telles "the four stages of the process are: planning, recording, editing and publishing." [2]

According to the TIC Education 2019 survey [3], 19% of the teachers who work in private schools use the podcast format to prepare their classes. Among those who teach in public schools, the percentage is 17%. The survey also points out that "podcasts are not a recent media resource on the Internet, [...] however, their appropriation by the public and professionals from the most diverse areas has recently intensified" [3].

The relationship of podcasts in Education is discussed by Bottentuit and Coutinho [4]. The authors address aspects of the potential use of this tool in teaching and learning processes, whether face-to-face or at a distance, in particular, highlighting that the podcast allows teachers to provide teaching materials in audio format that can be listened to at any time of day and anywhere.

Moura and Carvalho [5] point out the podcast as an educational medium and indicate that "the recording of classes [in podcast format] can help the teacher to manage and save time in classes whose contents do not change significantly from one year to the next". The authors also state that "talking about a podcast is talking about a class that can be studied or recalled at anytime, anywhere." Thus, one can reflect the appropriation of the tool in different pedagogical moments.

2. Review literature

We carried out a search on the Scopus database. By combining the terms "podcast" and "music education", we obtained four publications as a result. I present them below, in chronological order.

Among the works found, Gillies [6] highlights the impacts of technological evolution in relation to the different formats in which information can currently be archived, consulted and produced, thus altering spaces

* This work was funded by the Public Call n. 03/2020 Produtividade em Pesquisa PROPESQ/PRPG/UFPB
PVI13622-2020.

such as libraries and archive collections. While not discussing the podcast format per se, the author brings the term in his title as a provocation: “pen to podcast”. He points out such impacts in three contexts (university study, musical biography and musical analysis), and points out that such changes imply new forms of research and learning.

A pedagogical experience with podcasts in a Music Education class of elementary school in Portugal is presented by Coutinho and Mota [7]. Both teachers and students were involved in the process of creating content for a podcast. The authors emphasize that the “Web 2.0 offers educators an assortment of technologies and tools with enormous educational potential” [7] in addition to offering the opportunity for students to produce and publish content on the Internet. Furthermore, they indicate that podcasts have adequate characteristics for musical pedagogical practices, especially with regard to digital training. After analyzing the data, the authors indicate that once the Music Education context has an emphasis on audio, the production of podcasts in the classroom is relevant. Through the developed practices, students were able to develop musicality, technical and artistic skills, in addition to understanding the relationship between music and society, history, culture, different artistic languages and other areas of knowledge. At the end of the productions, the participants used positive adjectives to describe the experience performed. Among the difficulties found and reported, they highlighted the technical characteristics of the software used, such as the audio treatment and volume equalization of captures and recordings. Finally, the researchers point out that the “mere use” of podcasts can positively influence student learning and that the tool has potential for the area of Music Education.

Bolden and Nahachewsky [8] developed a pedagogical practice of creating podcasts with students from a music degree course in Canada. The authors highlight the concept of participatory culture in the sense that the students have the opportunity to create, produce and share podcasts, and not just consume them. They analyzed the perception of the students involved in the creative process and identified that once engaged in the tasks of creating a podcast, they developed skills to work with audio editing software, and those of self-knowledge and self-expression. The students also shared their productions, thus being able to celebrate their own achievements in a collaborative way and connect with others, thus developing meaningful personal connections. The authors concluded that it was a significant and effective experience and indicate that the production of podcasts can also be used in activities where there would normally be written work, such as the discussion of curriculum content, teaching experiences, vocal and instrumental practices, results of research, among so many other possibilities. They suggest that there is potential for the construction and representation of knowledge through the production of podcasts.

In another account of the same experience, Bolden and Nahachewsky [9] discuss the potential of creating podcasts as a methodology for exploring oneself. They collected data through interviews with students and the

participants were asked to create a three-to-five-minute podcast about their own story of, with and through music. There was a combination of narratives and music to describe and reflect personal experiences.

The authors discuss themes that emerged from the data: the integration between music and personal narratives that enhance the possibility of reflections, the speech previously written in contrast to the one actually spoken at the time of recording, listening to oneself through the repetition necessary for technical adjustments, enabling self-understanding, refining and developing ideas, the “personal” factor, in the sense of taking ownership as one's own creation, and the artistic and creative possibilities. A point highlighted by Bolden and Nahachewsky [9] is the interdisciplinarity in the practice of creating podcasts, since there is a combination of written text, narratives, different oralities, music, audio editing, sharing and social interaction, among so many possibilities in the context of Music Education.

Through the publications raised in the Scopus database, timid efforts can be seen in an emerging field. There is a report about elementary school and also about Higher Education. Although there is an experience developed with music teachers in training, the focus was not on their training process itself, in addition to the training pedagogical practice aimed at such students. Thus, there is a starting point in the works of Bolden and Nahachewsky [8] [9] that leads us to reflect on this scenario in the Brazilian context, also contemplating the pedagogical process itself developed through the curricular components of a music degree course.

It is also necessary to be aware of the quality of content production in the podcast format. In this sense, we highlight the work of Bottentuit Júnior and Coutinho [10]. The authors suggest recommendations on the production process in general, the initial and final moments of recording, the content, and group work. And they point out that “it is good to know how to proceed to create a good episode, which implies familiarizing yourself with technical issues, with aspects related to communication and, most importantly, with the suitability of the content provided”, leading us to reflect and propose quality podcast content production practices in the Music Education scenario.

Barros and Menta [11] reinforce that podcasts can help access music produced independently, and we add that several music students can take advantage of podcasts to release their digital albums and lessons. Thus, the authors emphasize that “some PodCasters, as PodCast producers are called, use music in their programs [...], in some cases entire programs are dedicated to the theme, a way to disseminate the productions that do not always have access to formal and institutional spaces for dissemination” [11].

In the context of a degree in music, when dealing with experiences involving a virtual learning environment and collaborative strategies in the development of a curricular component, Araldi [12] discusses an experience with the creation of podcasts by students. This activity was highlighted by the students as the one that best enabled the collective creation and the engagement among the undergraduates in the creation of a podcast.

It is clear that the exploration of podcasts as a format of pedagogical content still lacks further development and research in order to demonstrate the particularities of the format in music teacher training processes, besides the experimentation of producing digital audio content by the area of Music Education in Brazil.

3. Research design

General objective: Analyze the experience of producing digital pedagogical-musical content in podcast format by teachers and students in two Music Degree courses. And specific: Characterize the concept of podcast related to methodological specificities for the production of musical content; Provide a space for content production with music teachers and students; Identify the demands for pedagogical content to be developed in the podcast format; Produce pedagogical-musical material in audio with a focus on the context of primary and secondary schools levels; Analyze the process of producing podcasts by those involved.

In view of the objective of this research, the study is qualitative in nature and the methodology applied is based on the theoretical-methodological concepts of action research. Action research is a form of self-reflective investigation, carried out by participants in social situations in order to improve the rationality and justice of their own practices, the understanding of these practices and the situations in which they take place [13]. The authors reveal that the three aspects – the practices, the understanding of the practices and the conditions that shape them – are inevitably and incessantly linked together by unstable and volatile ties, continuously shaping each other.

In the educational field and also in music education, action research has been presented as a methodological strategy of great potential as it promotes the improvement of the professional teaching practice and the sharing of knowledge which can be useful in similar or different situations [14]. Therefore, we chose the practice of action research because we understand it is the most appropriate in relation to the object of study: the experience of producing pedagogical-musical content in the context of music teacher education.

The research is being developed by a team of faculty members of the Technologies and Music Education Research Group (Tedum-UFPB) linked to two federal educational institutions: Federal University of Paraíba (UFPB) and Federal Institute of Sertão Pernambucano (IF Sertão PE). The teaching team is responsible for selecting the participating students and planning the activities. Participating students are being selected from institutions that offer degree courses in music (UFPB; IF Sertão PE) and the selection criteria are: i) to be regularly enrolled in a degree course in Music at UFPB and IF Sertão PE; ii) have some previous experience with the use of digital technologies; iii) availability to develop experience in podcast creation.

Data collection will be carried out through the development of field diaries by the research team and

through conversation circles with the students participating in the research. In addition to the entire process of documentation, registration and analysis of the phases that make up the action research cycle.

It is noteworthy that the involvement of the faculty and of undergraduate students from the music degree course in the podcast production processes implies an immediate return in their training and pedagogical praxis, benefiting them with the experience and immersion in digital practices that pay attention to contemporary particularities.

3.1 Hypothesis

Through this study, we expect to conclude that the production of pedagogical-musical content in the podcast format implies knowledge and contact with specific methodologies for this content format, allowing the participating undergraduates the opportunity to learn about content production for online and blended teaching, and for the teachers involved, a reflection on the potential of this practice.

4. Risks, benefits and outcomes

Considering the possibility of holding press conferences during face-to-face meetings, the students may present some discomfort in exposing their ideas and criticism regarding the work process. One way to alleviate this discomfort can be a mix of group and individual interviews.

The possibility of participating in the research indicating the lessons learned, doubts and suggestions about improvements in the production of these pedagogical resources in audio/podcasts allows participants to expand the possibilities of digital content production for courses that can be used both in remote and on-site teaching.

We expect that from the analysis of the perception of students and teachers participating in the research, it will be possible to understand specific practical and methodological aspects in the development of pedagogical-musical podcasts, thus contributing to the implementation of this format in the performance of these undergraduate students in primary and secondary schools levels. The research process will also enable a first systematization about the contributions of creating digital pedagogical-musical content.

6. References

- [1] Barros, M. H. da F. Educação musical, tecnologias e pandemia: reflexões e sugestões para o ensino remoto emergencial de música. *OuvirOUver*, v. 16, nº 1, 2020, p. 292-304.
- [2] Telles, A. *A revolução das Mídias Sociais: Cases, Conceitos, Dicas e Ferramentas*. 2º ed. São Paulo: M. Book, 2011.
- [3] Cetic.BR. *Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras: TIC Educação* 2019. 1º ed. São Paulo: Comitê Gestor da

Internet no Brasil, 2020.

[4] Bottentuit Junior, J. B.; Coutinho, C. P. Podcast em educação: um contributo para o estado da arte. In: *Libro de Actas do Congreso Internacional Galego-Portugués de Psicopedagogía. Universidade da Coruña*. 2007. p. 837-846.

[5] Moura, A. M. C.; Carvalho, A. A. A. Podcast: uma ferramenta para usar dentro e fora da sala de aula. In: Conference on Mobile and Ubiquitous Systems. *Anais. Universidade do Minho: Braga*, 2006, p. 155-158.

[6] Gillies, M. From pencil to podcast: maximizing musical resources. *Fontes Artis Musicae*. v. 55, nº 1, 2008. p. 101-104.

[7] Coutinho, C.; Mota, P. web 2.0 Technologies in Music Education in Portugal: Using Podcasts for Learning. *Computers in the Schools: Interdisciplinary Journal of Practice, Theory, and Applied Research*, v. 28, nº 1, p.56-74, 2011. DOI: <https://doi.org/10.1080/07380569.2011.552043>

[8] Bolden, B.; Nahachewsky, J.. Podcast creation as transformative music engagement. *Music Education Research*, v. 17, nº 1, p. 17-33. 2015a. DOI: <https://doi.org/10.1080/14613808.2014.969219>

[9] Bolden, B.; Nahachewsky, J. Podcast creation: a methodology for exploring self within music teacher education. *Journal of Music, Technology & Education*, v. 8, nº 3, p. 243-260, 2015b. DOI: https://doi.org/10.1386/jmte.8.3.243_1

[10] Bottentuit Junior, J. B.; Coutinho, C. P.. Recomendações para Produção de Podcasts e Vantagens na Utilização em Ambientes Virtuais de Aprendizagem. *Revista Prisma*, n. 6, p.158-179, 2008. Disponível em: <http://ojs.letras.up.pt/index.php/prismacom/article/view/3217>

[11] Barros, G. C.; Menta, E.. Podcast: produção de audio para educação de forma crítica, criativa e cidadã. In: *Revista de Economía Política de las Tecnologías de la Información y Comunicación*, vol. IX, n. 1. Abril, 2007.

[12] Araldi, J. O papel das tecnologias digitais na formação do professor de música. In: Congresso Nacional da Associação Brasileira de Educação Musical, 21, *Anais. Pirenópolis, GO*. p. 1700-1709.

[13] Carr, W.; Kemmis, S. *Becoming Critical: education, knowledge and action research*. London and Philadelphia: The Palmer Press, 1986.

[14] Noffke, S. E. Professional, personal, and political dimensions of action research. *Review of Research in Education*, v. 22, n. 1, 1997, p. 305–343.

Song Emotion Recognition: A Study of the State of the Art

Arthur Nicholas dos Santos^{1*}, Karen Gissell Rosero Jácome^{1†}, Bruno Sanches Masiero¹.

¹Laboratório de Acústica das Comunicações (LAC) – Faculdade de Engenharia Elétrica e de Computação (FEEC) da Universidade Estadual de Campinas (UNICAMP)
Av. Albert Einstein, 400, Bloco A – 13083-852 Campinas, SP

a264372@dac.unicamp.br, k264373@dac.unicamp.br, masiero@unicamp.br

Abstract. *Music is art, and art is a form of expression. Often, when a song is composed or performed, there may be an intent by the singer/songwriter of expressing some feeling or emotion through it, and, by the time the music gets in touch with an audience, a spectrum of emotional reactions can be provoked. For humans, matching the intended emotion in a musical composition or performance with the subjective perceptiveness of different listeners can be quite challenging, in account that this process is highly intertwined with people's life experiences and cognitive capacities. Fortunately, the machine learning approach for this problem is simpler. Usually, it takes a data-set, from which features are extracted to present this data to a model, that will train to predict the highest probability of an input matching a target. In this paper, we studied the most common features and models used in recent publications to tackle music emotion recognition, revealing which ones are best suited for songs (particularly acapella).*

1. Introduction

Music Emotion Recognition (MER) is a sub-field of Music Information Retrieval (MIR) that deals with classification of music according to affection [1]. The importance of MER can be justified by the dependency that search and recommendation engines have on metadata. In simple terms, metadata is data about data. For example, when a person takes a picture of a cat using a smartphone, the picture is the data itself. However, a series of other information about that data is also recorded, e. g., the time and date when the picture was taken, or the geographical coordinates where it was captured. These are all secondary information about a primary one, that can be used to tag it, in order to retrieve it in the future, as well as finding other information that is similar to it.

Song metadata like genre, composer, artist, album, year of release etc. are commonly used by streaming services to help users find what they like, and even recommend songs they might like based on their listening history. However, the mood of a song can also be considered an interesting metadata that could be used to relate a certain song to similar content.

This paper studies various recent articles published on MER, including papers on song and instrumental music emotion recognition, and is organized as follows: Section 2 details our findings on what are the most commonly used features for representing music samples in order to present this type of data to a machine learning (ML)

model, and covers our findings on the most commonly used model architectures. Section 3 describes our experiment, which consists in creating a couple of ML models based on the information retrieved on the previous sections, as well as the data-set used to test our models. Section 4 shows our results in terms of accuracy, comparing it to previous works that used the same data-set. Finally, Section 5 draws some conclusions on what are the best combination of features and models for song emotion recognition that we found.

2. Features and Models

According to Panda, Malheiro and Paiva, musical dimensions can be related to emotions by a set of high-level features, namely: melody, harmony, rhythm, dynamics, tone color (timbre), expressivity, texture, form, and vocals. On the other hand, computational features are considered low-level, because they only provide primitive descriptions by which individual high-level features may be identified [2].

By reviewing 10 articles on MER published in 2020 alone we have found 47 different low-level computational features being used separately or concatenated, to represent different aspects of the aforementioned high-level features [3–12]. All these features are available off-the-shelf on Python libraries and MATLAB toolboxes, and 6 of them were found to be used on 76.6% of the publications reviewed:

Spectral roll-off: relates to tone color and indicates the frequency below which approximately 85% of the magnitude spectrum distribution is concentrated [2]. Was used in [3], [5], [8–10], and [12].

Zero-crossing rate (ZCR): also relates to tone color and represents the number of times a waveform changes sign in a window, indicating change of frequency and noisiness [2]. Was used in [3], [5], [8–10], and [12].

Spectral centroid: also relates to tone color and represents the mean of the magnitude spectrum of the short-time Fourier Transform (STFT) [2]. Was used in [3–5], [8, 9], and [12].

Mel spectrogram: also relates to tone color and decomposes an audio signal into a series of frequency channels inspired by the human cochlea, enabling to study the signal's frequency distribution into so-called critical bands [2]. Was used in [4–8], and [11].

Mel-frequency Cepstral Coefficients (MFCC): also relates to tone color and measures spectral shape. Can be derived from a log magnitude Mel spectrogram based

*Grant #2019/22795-1, São Paulo Research Foundation FAPESP.

†Grant #2019/22945-3, São Paulo Research Foundation FAPESP.

on the Discrete Fourier Transform (DCT). Typically, only the first 8 to 13 MFCCs are used for speech representation [2]. Was used in [3–5], [9, 10], and [12].

Chromagram: relates to harmony and indicates energy distribution along pitch classes in a 12-dimensional vector (12 semitones, from A to G#) [2]. Was used in [4, 5], [8–10], and [12].

As for the other 41 features found (which were used on only 23.4% of the publication reviewed), 13 of them are related to rhythm, 10 are related to tone color, 6 are related to harmony, 5 are related to melody and dynamics (each), and only 1 is related to texture, form and vocals (each).

Depending on the architecture of the ML model used as a classifier for MER, several of the aforementioned front-end features can be used together, to better represent the training data. However, not all models allow that. According to de Azevedo and Bressan, what dictates which and how many features can be used as front-end for an ML model is the architecture of the model itself [3]. In our study, 12 different architectures were found to be used, separately or combined, and 3 of them were found to be used on 17% of the publications reviewed (each):

Support Vector Machine (SVM): is a binary classifier that divides the training data into groups by using hyper-planes. An SVM finds an optimal hyper-plane by using the dot product functions in feature space using kernel functions. The solution of the optimal hyper-plane can be written as a combination of a few input points that are called support vectors [3]. Was used in [3], [6], [8, 9] and [12].

Multi-Layer Perceptron (MLP): is an artificial neural network (ANN) that models the relationship between a set of training data and known targets. Its architecture is based on a simplified understanding of how the human brain responds to stimuli from sensory organs and is best suited to problems where the relation between input and output data is well understood, yet the process that relates both is extremely complex [3]. Was used in [3], [5], [8, 9] and [12].

Convolutional Neural Network (CNN): is a type of ANN based on convolutional operations that can extract high-level features from 2-dimensional low-level ones, such as Mel-spectrogram, MFCCs and chromagram. It deeply extracts underlying features contained in each time frame, while retaining time-series features in the same direction. After a CNN block, a fully connected MLP block is often used to predict outputs in classification problems [4]. Was used in [4], [6, 7], and [11, 12].

As for the other 9 architectures, recurrent neural networks (RNN) with long-term short-memory (LSTM) blocks and random forest were found to be used on 10% of the publications reviewed (each), k-nearest neighbors was found to be used on 7% of the publications reviewed, RNN with gated recurrent unit (GRU) blocks, decision tree (CART and C4.5) and state vector regressor were found to

be used on 4% of the publications reviewed (each) and logistic regression was found to be used on 3% of the publications reviewed.

3. Experimental Setup

To experiment with the features and models described in Section 2, a portion of the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) data-set was chosen. This portion comprises 1,012 audio-only files of song recordings, performed by 23 actors singing 2 lexically matched statements in a neutral North American accent. Song emotions include neutral, calm, happy, sad, angry, and fearful expressions [13].

Since an MLP model can have an input layer with as many neurons as necessary, all input features can be concatenated and flattened into a 1-dimensional input vector. Therefore, we used the principal component analysis (PCA) technique to reduce the dimensionality of each of the 6 features presented in Section 2, aiming to visualize the minimum number of variables that keeps the maximum amount of information about how the feature data is distributed, as illustrated in Figure 2. Mel spectrogram was discarded because it showed the worst clustering and the highest dimensionality compared to the other features, moreover we included MFCCs that are already computed based on it. Hence, we chose to train an MLP model using the concatenation of the 5 selected features.

Table 1 summarizes the architecture of our MLP model, which has an input layer with 11,394 neurons, followed by 2 hidden layers with 1,024 and 128 neurons, respectively, totaling 141,635,084 trainable parameters. ReLU was used as an activation function for all layers, except for the output layer, where Softmax was implemented. For regularization, Dropout was used, which randomly ignores a percentage of neurons during training.

Table 1: Summary of our MLP model

Layer type	Output shape	Param. #
Dense	(None, 11,394)	129,834,630
Dropout	(None, 11,394)	0
Dense	(None, 1,024)	11,668,480
Dropout	(None, 1,024)	0
Dense	(None, 128)	131,200
Dropout	(None, 128)	0
Dense	(None, 6)	774

For our CNN model, we selected the 2-dimensional feature that showed the clearest clustering in the PCA visualization (Figure 2), which was the chromagram. Table 2 summarizes the architecture of our CNN model, which has an input shape of (12, 422, 1). To obtain the output for the second convolutional layer, 24 kernels with a shape of (5, 5) were used, with a stride of (1, 1). Another typical CNN operation, which is Pooling, was used to down-sample data and compress information, with a shape of (2, 4) and a stride of (2, 4). The third convolutional layer used 48 kernels with a shape of (2, 2), followed by a Pooling operation, with a shape of (1, 3) and

a stride of (1, 3). After the last convolutional layer, the data was flattened into a 1-dimensional vector with 1,536 elements, to be fed into 2 fully connected layers, with 64 and 6 neurons, respectively, totaling 124,822 trainable parameters. Dropout was used for regularization and ReLU as activation function for all layers, except for the output layer, where Softmax was used.

Table 2: Summary of our CNN model

Layer type	Output shape	Param. #
Conv2D	(None, 8, 418, 24)	624
MaxPooling2D	(None, 4, 104, 24)	0
Activation	(None, 4, 104, 24)	0
Conv2D	(None, 3, 103, 48)	4,656
MaxPooling2D	(None, 3, 34, 48)	0
Activation	(None, 3, 34, 48)	0
Conv2D	(None, 1, 32, 48)	20,784
Activation	(None, 1, 32, 48)	0
Flatten	(None, 1,536)	0
Dropout	(None, 1,536)	0
Dense	(None, 64)	98,368
Activation	(None, 64)	0
Dropout	(None, 64)	0
Dense	(None, 6)	390
Activation	(None, 6)	0

Since we were dealing with a multi-classification problem, and an SVM is a binary classifier, we opted to focus on the two architectures already discussed (MLP and CNN) and do not implement an SVM model.

4. Results and Discussion

To train both our models, the data-set was split into 622 samples for training, 200 for validation and 200 for test. Both our models were trained for 100 epochs, however, Figure 1 illustrates that the MLP learning curves begin to diverge at around epochs 50-55, while the CNN learning curves begin to diverge at around epochs 25-30. This shows that the CNN model achieve its best results in less training epochs, compared to the MLP model.

On the left hand side of Figure 2, PCA plots are illustrated for the concatenated features (top) and the chromagram alone (bottom), both before training, exposing mixed clusters. On the right hand side, it is shown that the samples are nicely spaced and grouped together after the classification performed by our MLP (top) and CNN (bottom) models. This evinces that the use of ML architectures together with a good feature selection is a favorable technique for MER.

The overall test accuracy of the MLP model achieved only 67.7%, while the CNN achieved 80.6%. Moreover, Figure 4 shows that the MLP test accuracy per class ranges from 6.8% (happy) to 16.7% (angry), while the CNN test accuracy per class has a worst-case scenario of 11.2% (happy) and best-case scenario of 16.7% (fearful).

We resorted to the works of [14–16] to compare our results with papers that used the same data-set, but not

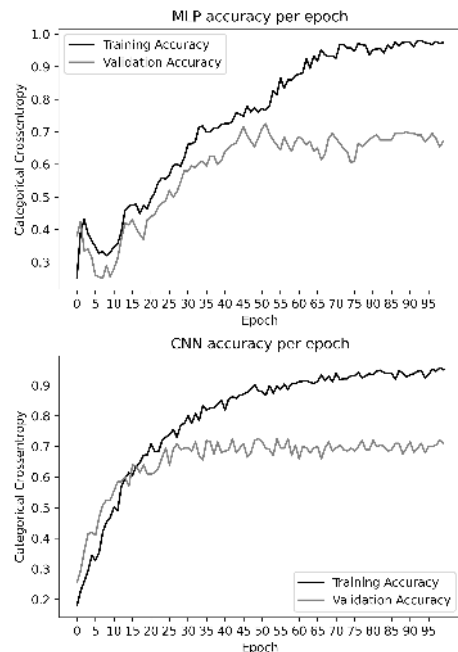


Figure 1: MLP and CNN learning curves

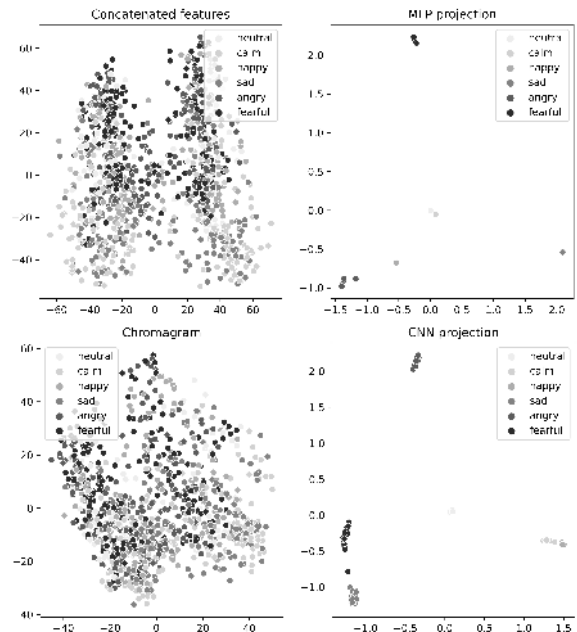


Figure 2: PCA visualization for MLP and CNN

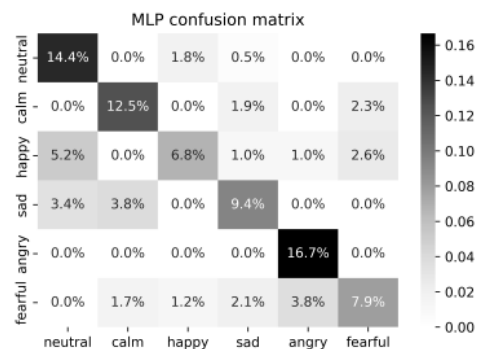


Figure 3: MLP confusion matrix using 5 features

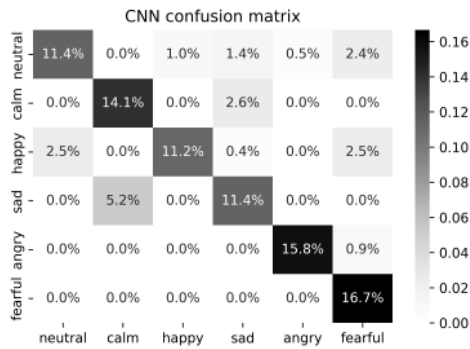


Figure 4: CNN confusion matrix using 1 feature

necessarily the same features and model architectures. Table 3 summarizes each work’s authors, features and architectures used, as well as the overall test accuracy obtained.

Table 3: Accuracy comparison with other works

Authors	Features	Models	Acc.
[14]	MATLAB Audio Analysis Library	Logistic Regression	48%
Ours	Concatenated features	MLP	67.7%
Ours	Chromagram	2D CNN	80.6%
[16]	LibROSA HSF	RNN (LSTM)	82%
[15]	MFCC	1D CNN +BiLSTM	94%

We also performed a forensic analysis on our MLP model, to see which features yield the best performance. Using only the chromagram, the overall accuracy achieved was equal to 66.7%, while using only MFCC it was 45.9%. Spectral roll-off, ZCR and spectral centroid yielded the worst performances, with 29.6%, 28,5% and 24.7% overall test accuracy, respectively.

5. Conclusions

Although the most popular computational features for MER used in recent publications relate to tone color, the chromagram, which relates to harmony, was found to be best suited for song emotion recognition, in our experiments. Also, our CNN model performed better than our MLP model, both in terms of overall test accuracy and accuracy per-class, because our MLP model favored 3 emotions (neutral, calm and angry), while causing more confusions for the predictions of the other 3 emotions (happy, sad and fearful). Finally, it was evinced by comparison with the works of [15, 16] that CNN performance can be further improved with the use of RNN blocks, that can learn order dependence in sequence prediction problems, which may be a desired behavior in song emotion recognition.

References

[1] Youngmoo E Kim, Erik M Schmidt, Raymond Migneco, Brandon G Morton, Patrick Richardson, Jeffrey Scott,

Jacquelin A Speck, and Douglas Turnbull. Music emotion recognition: A state of the art review. In *Proc. ismir*, volume 86, pages 937–952, 2010.

- [2] Renato Panda, Ricardo Manuel Malheiro, and Rui Pedro Paiva. Audio features for music emotion recognition: a survey. *IEEE Transactions on Affective Computing*, pages 1–1, 2020.
- [3] Beatriz Flámia Azevedo and Glaucia Bressan. A comparison of classifiers for musical genres classification and music emotion recognition. pages 241–262, January 2018.
- [4] Zijing Gao, Lichen Qiu, Peng Qi, and Yan Sun. A novel music emotion recognition model for scratch-generated music. In *2020 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1794–1799, June 2020.
- [5] Laugs Casper. Creating a speech and music emotion recognition system for mixed source audio. Master’s thesis, August 2020.
- [6] Mladen Russo, Luka Kraljević, Maja Stella, and Marjan Sikora. Cochleogram-based approach for detecting perceived emotions in music. *Information Processing & Management*, 57(5):102270, September 2020.
- [7] Wooyeon Kim. Musemo: Express musical emotion based on neural network. Master’s thesis, February 2020.
- [8] Ana Gabriela Pandrea, Juan Sebastián Gómez-Cañón, and Perfecto Herrera. Cross-Dataset Music Emotion Recognition: an End-to-End Approach, 2020.
- [9] Pengfei Du, Xiaoyong Li, and Yali Gao. Dynamic music emotion recognition based on cnn-bilstm. In *2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC)*, pages 1372–1376, 2020.
- [10] Yesid Ospitia Medina, José Ramón Beltrán Blázquez, and Sandra Baldassarri. Emotional classification of music using neural networks with the mediaeval dataset. *Personal and Ubiquitous Computing*, April 2020.
- [11] Sangeetha Rajesh and N J Nalini. Musical instrument emotion recognition using deep recurrent neural network. *Procedia Computer Science*, 167:16–25, 2020. International Conference on Computational Intelligence and Data Science.
- [12] Stuart Cunningham, Harrison Ridley, Jonathan Weinel, and Richard Picking. Supervised machine learning for audio emotion recognition. *Personal and Ubiquitous Computing*, April 2020.
- [13] Steven R. Livingstone and Frank A. Russo. The ryer-son audio-visual database of emotional speech and song (ravdess): A dynamic, multimodal set of facial and vocal expressions in north american english. *PLoS ONE*, 13(5), 2018.
- [14] Stuart Cunningham, Jonathan Weinel, and Richard Picking. High-level analysis of audio features for identifying emotional valence in human singing. In *In: Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, pages 1–4, September 2018.
- [15] Ashima Yadav and Dinesh Kumar Vishwakarma. A multilingual framework of cnn and bi-lstm for emotion classification. In *2020 11th International Conference on Computing, Communication and Networking Technologies (IC-CCNT)*, pages 1–6, 2020.
- [16] Bagus Tris Atmaja and Masato Akagi. On the differences between song and speech emotion recognition: Effect of feature sets, feature types, and classifiers. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pages 968–972, 2020.

Análise de características acústicas para identificação de *triggers* em gêneros e áudios musicais distintos

Marco Antonio dos Santos Fernandes, Estela Ribeiro, Carlos Eduardo Thomaz

¹ FEI, departamento de Engenharia Elétrica

Av. Humberto de Alencar Castelo Branco, 3972 – B – 09850-901, São Berardo do Campo, SP

marcoantoniosf2207@gmail.com, estela.eng@hotmail.com, cet@fei.edu.br

Resumo

Este trabalho analisa em detalhes uma base pública internacional de áudios musicais acerca das características acústicas presentes nas amostras existentes. Ao todo mil áudios classificados em dez gêneros musicais foram analisados em relação às suas características acústicas, com objetivo de buscar padrões nos *triggers* presentes nas amostras de cada gênero musical e a quantidade destes para aplicação em estudos subsequentes de ativações cerebrais geradas pela escuta destas músicas. Os resultados mostram que dentre os dez gêneros musicais presentes na base de áudios, dois desses (*Disco* e *Metal*) não apresentam número de *triggers* suficiente para tal aplicação.

1. Introdução

A música é complexa e possui vários gêneros musicais, sendo usada em diferentes situações. Em um festival, show ou reunião entre amigos a música se faz presente, podendo até ser o componente essencial para que o evento aconteça [1]. Ao escolher o tipo de música que gostamos, realizamos implicitamente julgamentos que correspondem às nossas preferências musicais [2, 3].

Um áudio musical possui diversas características [4] que são estímulos para o cérebro, tais como harmonia, melodia e ritmo, gerando ativações cerebrais. Estudos recentes utilizando estímulos sonoros [5, 6, 7, 8] surgiram para compreender as respostas neurológicas geradas durante a escuta musical em instantes denominados de *triggers* [5,6].

Este trabalho estende as metodologias de identificação de *triggers* [5] e de análise das características acústicas [6, 7, 8] em uma base de áudios composta por trechos de músicas de gêneros variados para averiguar a presença de *triggers* nestas amostras e verificar a possibilidade de utilização desses áudios em estudos de ativações cerebrais pertinentes.

2. Metodologia

A metodologia pode ser dividida em três etapas: (I) Extração de características acústicas; (II) Extração de

triggers; (III) Análise dos *triggers*.

Para primeira etapa, utiliza-se a base de áudios pública e internacionalmente conhecida GTZAN [4] que contém mil trechos musicais de trinta segundos de duração, contendo dez gêneros musicais com cem músicas cada. Os gêneros musicais estudados são: (1) *Blues*, (2) *Música Clássica*, (3) *Country*, (4) *Disco*, (5) *Hip-Hop*, (6) *Jazz*, (7) *Metal*, (8) *Pop*, (9) *Reggae* e (10) *Rock*.

Destes áudios foram extraídas características acústicas de baixo nível [9, 10] capazes de gerar respostas neurais significativas [5, 6]. As características extraídas são: (1) *Root Mean Square Energy* (RMS), (2) *Zero Crossing Rate* (ZCR), (3) *Brightness*, (4) *Spectral Centroid*, (5) *Spectral Spread*, (6) *Spectral Flatness*, (7) *Spectral Skewness*, (8) *Spectral Kurtosis*, (9) *Spectral Rolloff*, (10) *Spectral Flux*, (11) *Spectral Roughness* e (12) *Spectral Entropy*. A ferramenta utilizada para extração das características foi a MIRtoolbox (versão 1.7.2) [11], no software Matlab R19a. Os sinais das características exploradas foram decompostos em janelas de 50 ms de duração com 50% de sobreposição [6, 7, 8].

Na segunda etapa, são extraídos os *triggers*, que são instantes no áudio em que há estímulo capaz de gerar respostas neurais significativas (ou potencial evocado) e ocorrem por mudanças acentuadas nas características acústicas [5]. Segundo a metodologia de Poikonen et al. (2016) [5] de identificação de *triggers* é necessário determinar os limiares inferiores e superiores (V_{inf} e V_{sup}) que são uma porcentagem do sinal que está acima e abaixo da sua média, definida aqui como 20%. Quando o sinal permanece abaixo de V_{inf} por um período mínimo chamado *Preceding Low-Feature Phase* (PLFP) e em seguida ocorre um crescimento rápido, ultrapassando o limiar superior V_{sup} chamado de Magnitude de Rápido Crescimento (MoRI), há a ocorrência de um *trigger* ao final desta sequência, como mostra a Figura 1. Os valores de PLFP e MoRI são de 500 ms e 150 ms, respectivamente [6, 8].

Na análise dos *triggers*, terceira e última etapa, é realizada a detecção de padrões entre a presença de *triggers* nos diferentes gêneros, estendendo a metodologia de Ferreira et al. (2019) [7] e Ribeiro et al. (2020, 2019) [6,8] das características acústicas, obtendo a quantidade de *triggers* por amostra de áudio, permitindo a sua avaliação quantitativa.

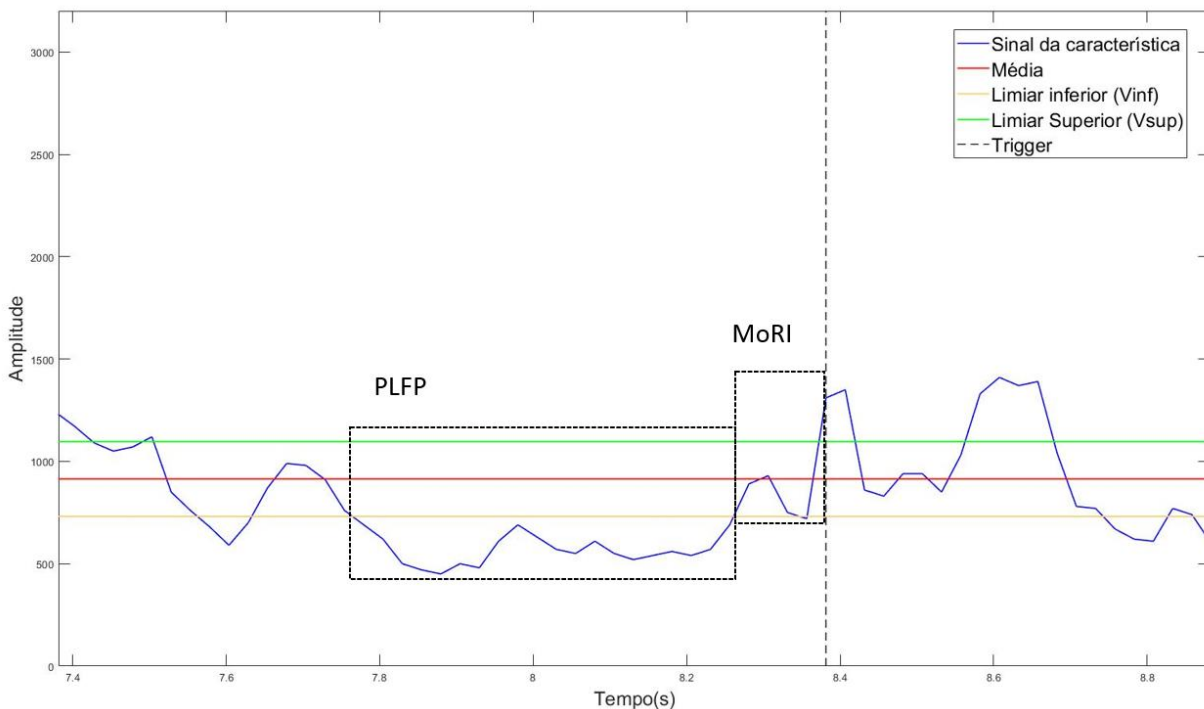


Figura 1: Exemplo de *trigger*.

3. Resultados

Há três agrupamentos entre as características. O primeiro agrupamento é composto por RMS, *Roughness* e *Spectral Flux*; o segundo por *Skewness* e *Kurtosis*; O terceiro por ZCR, *Spectral Rolloff*, *Brightness*, *Spectral Entropy*, *Spectral Flatness*, *Spectral Centroid* e *Spectral Spread*.

Os agrupamentos foram encontrados através de Análise Fatorial que indicou redundância entre as características acústicas, então foram selecionadas as características que possuíam maior carga fatorial dentro de cada agrupamento, sendo essas: RMS, *Spectral Rolloff* e *Spectral Skewness*. Devido ao fato de cada amostra de áudio possuir apenas trinta segundos de duração, estas amostras não apresentam grandes quantidades de *triggers*. A dispersão de *triggers* para estas características selecionadas é importante para que seja possível identificar a quantidade de músicas que podem ser usadas, conforme mostram as Figuras 2, 3 e 4.

A característica *Root Mean Square Energy* (RMS) possui poucas amostras válidas para os gêneros *Disco* e *Metal*. *Disco* possui apenas nove músicas com pelo menos dois *triggers* e duas músicas com um *trigger*, logo há poucas amostras que possuem instantes significativos para análise de resposta neural. O gênero *Metal* apresenta números um pouco melhores, sendo dez áudios com pelo menos dois *triggers*. Os demais gêneros musicais apresentam mais de vinte músicas com mais de dois *triggers*, logo oferecem quantidade mínima e suficiente de informação para estudos envolvendo respostas neurais. Alguns áudios apresentam muitos *triggers*, como é o caso de “blues.00026.wav” que possui quinze *triggers*, e áudios

com treze *triggers*, que são “blues.00011.wav” e “disco.00080.wav”, que é um ponto discrepante do gênero *Disco* que tem 89 áudios sem *triggers* RMS.

Spectral Rolloff foi a característica dentre as principais que apresentou menor número de *triggers*. Os gêneros *Disco* e *Metal* possuem dois e cinco áudios, respectivamente, que têm um *trigger*, apenas. Os outros gêneros apresentam pelo menos quinze áudios diferentes com no mínimo um *trigger*, e há casos de um único áudio com mais de dez *triggers* nos gêneros *Rock* e *Country*.

Todos os gêneros possuem ao menos dez áudios com pelo menos um *trigger* detectado pela característica *Spectral Skewness*. O gênero *Classical* foi o gênero com menos amostras, sendo apenas onze que possuem apenas um *trigger*.

Estes resultados mostram que apesar de haver muitas amostras na base de áudios GTZAN, muitos arquivos não possuem *triggers* suficientes. Portanto, a utilização desses áudios em estudos envolvendo análise de ativações cerebrais não seria possível.

4. Conclusão

Este trabalho apresenta os resultados do mapeamento da base de áudios GTZAN para identificar quais das suas amostras possuem *triggers* das principais características acústicas, identificando que apesar de haver muitos trechos de áudio na base, os gêneros *Disco* e *Metal* possuem pouquíssimas amostras significativas em relação a conteúdo de *triggers* para a característica *Spectral Rolloff* que é a característica acústica mais significativa do terceiro agrupamento, portanto é muito importante em relação ao conteúdo acústico do agrupamento. Este achado é

importante para utilização desta base de áudios em estudos de ativações cerebrais motivadas por estas músicas, porque

a partir destes resultados é possível aproveitar apenas os áudios que possuem as melhores quantidades de *triggers*.

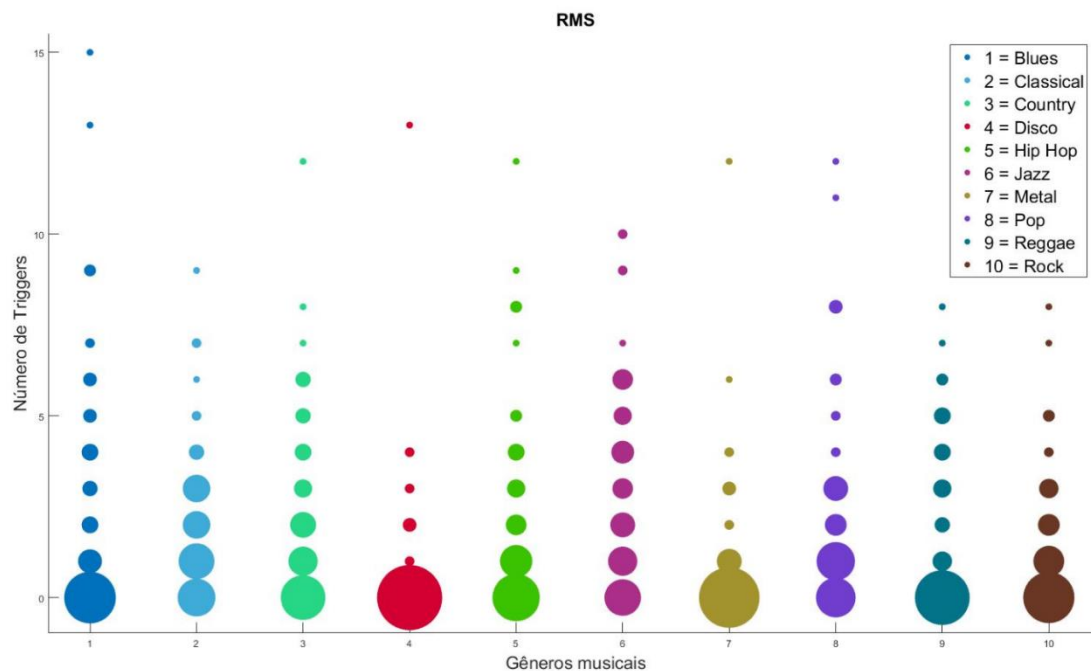


Figura 2: Distribuição de áudios por número de *triggers* – RMS.

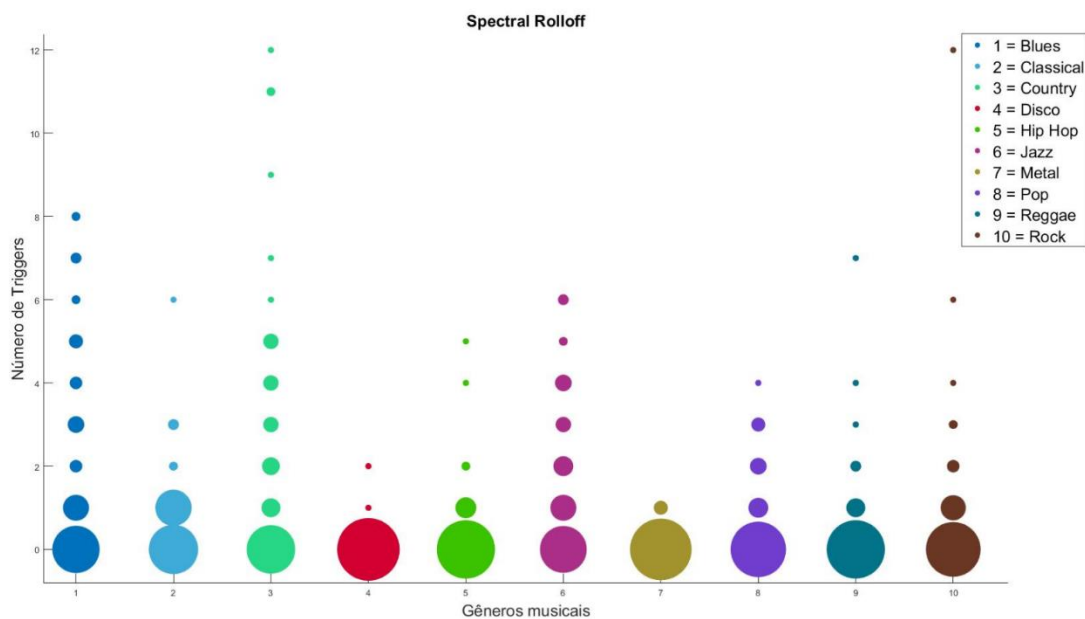


Figura 3: Distribuição de áudios por número de *triggers* – Spectral Rolloff.

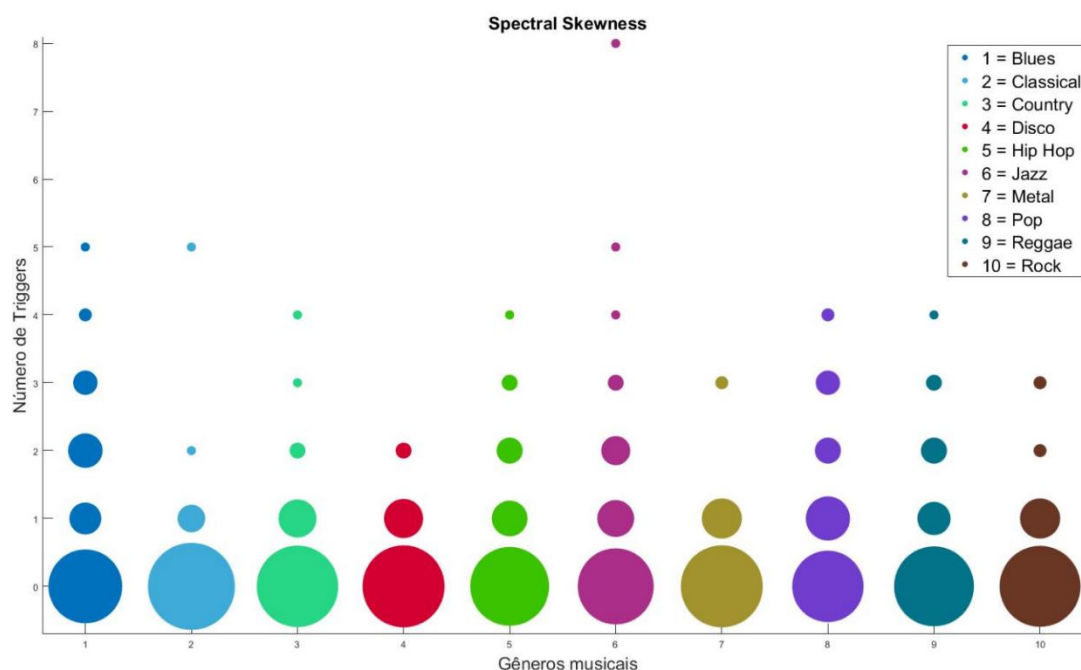


Figura 4: Distribuição de áudios por número de *triggers* – Spectral Skewness.

Agradecimentos

Os autores agradecem a bolsa FEI (PBIC006/20) concedida ao primeiro autor para realização desta pesquisa.

Referências

- [1] RENTFROW, Peter J.; GOSLING, Samuel D. The do re mi's of everyday life: The structure and personality correlates of music preference. *Journal of Personality and Social Psychology*, v. 84, n. 6, p. 1236-1256, 2003.
- [2] GREENBERG, David M.; et al. Musical Preferences are linked to cognitive styles. *PLOS ONE*, pp. 1-22, 2015.
- [3] SOLEYMANI Mohammad; et al. Content-Based music recommendation using underlying music preference structure. *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1-6, 2015.
- [4] TZANETAKIS, G.; COOK, P. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, v. 10, n. 5, p. 293–302, jul. 2002.
- [5] POIKONEN, H. et al. Event-related brain responses while listening to entire pieces of music. *Neuroscience*, v. 312, p. 58-73, jan. 2016.
- [6] RIBEIRO, Estela; THOMAZ, Carlos E. Whole brain EEG analysis of musicianship. *Music Perception*, v.37, pp. 42-56, 2019.
- [7] FERREIRA, Leonardo Antunes; RIBEIRO, Estela; THOMAZ, Carlos Eduardo. A cluster analysis of benchmark acoustic features on Brazilian music. In:

SBCM, pp. 1-3, 2019.

- [8] RIBEIRO, Estela. ANÁLISE E RECONHECIMENTO DE PADRÕES COGNITIVOS EM ESCUTAS MUSICAIS E SONOROS EM ÁUDIOS. 2020. Dissertação (Doutorado em Engenharia Elétrica) – Centro Universitário FEI, São Bernardo do Campo.

- [9] LERCH, Alexander. An introduction to audio content analysis. 1st ed. New Jersey: IEEE, 2012.

- [10] KNESS, Peter; SCHEDL, Markus. Music similarity and retrieval: an introduction to audio and web based strategies. 1st ed. Heidelberg: Springer, 2016.

- [11] LARTILLOT, Oliver. MIRtoolbox 1.7.2 users manual. Aalborg: Department of architecture, design e media technology, 2019.

Identificação de Áudio Vocaloid com *Support Vector Machines*: um Estudo de Caso da Hatsune Miku

Felipe V. de Almeida¹, Victor T. Hayashi¹

¹Escola Politécnica – Universidade de São Paulo (USP)

{felipe.valencia.almeida,victor.hayashi}@usp.br

Resumo. O processamento de sinais de áudio em conjunto com modelos de aprendizado de máquina tem aplicações em diversas áreas: música, análise forense e análise de fala humana e ruído ambiente. Analisar música de diferentes gêneros pode encorajar investigações interessantes pela comunidade científica, como a investigação de tendências culturais. Este trabalho apresenta uma iniciativa para o processamento de canções vocaloides que ganharam grande popularidade nas redes sociais. Os classificadores *Support Vector Machine (SVM)* foram treinados em dois experimentos para distinguir canções da vocaloide Hatsune Miku de canções instrumentais e canções de outros vocaloides, apresentando resultados promissores de precisão acima de 80%, que validam a iniciativa.

1. Introdução

A área de pesquisa de processamento de áudio usando modelos de aprendizado de máquina possui grande interesse da comunidade científica. Há exemplos na literatura em diversos cenários como o monitoramento não invasivo de colônias de abelhas [1], categorização de músicas [2], análise forense [3], além de análises de fala e ruídos ambientais [4].

Por exemplo, o desafio ASVspoof 2017 [5] consistiu em um desafio da área de reconhecimento automático de locutor para obter um classificador que pudesse diferenciar sons provenientes de humanos de sons provenientes de outros equipamentos. Este classificador pode ser usado para mitigar ataques de *replay* físico realizados a partir de um dispositivo próximo à interface de voz.

O processo de extração de informações de músicas necessita de bases de dados representativas e que estejam de acordo com o escopo de trabalho definido [6]. Esforços de pesquisa relevantes foram encontrados para criação de bases de dados de músicas árabe-andaluzas a partir de escolas de música no Marrocos [7] e de músicas indianas [8]. O estudo das músicas em uma perspectiva histórica fomenta também um melhor entendimento da cultura e gostos de diferentes comunidades.

Vocaloid pode ser definido como um sintetizador vocal desenvolvido originalmente pela empresa *Yamaha Corporation*, sendo um dos primeiros recursos deste tipo para uso da comunidade [9]. Inicialmente o uso deste recurso era limitado para a comunidade oriental, tendo pouca difusão no ocidente, porém isso mudou com o advento das VTubers (*Virtual Youtubers*) que tiveram boa aceitação no ocidente principalmente pela participação em

canais como Youtube, Twitch, Tik Tok, entre outros [10]. Tanto o vocaloid quanto a VTuber além de possuírem o recuso de sintetização vocal costumam acompanhar uma representação tridimensional de uma personagem. No caso das VTubers, pode-se citar a Kizuna AI como a primeira e mais famosa [11], e já no caso dos vocaloides destaca-se a Hatsune Miku, que é o estudo de caso deste trabalho. A importância da representação tridimensional de uma personagem está vinculada a uma identidade visual e um apelo ao público [12], o que reflete no sucesso dos casos como apontado por sites como NicoNico [13] e BiliBili [14].

Hatsune Miku, ilustrada pela Figura 1, é a vocaloid mais conhecida, tendo gerado diversos produtos comerciais, desde shows com representação holográfica até jogos e miniaturas. Criada em 2007 pela *Crypton Future Media*, seu surgimento impulsionou o uso dos vocaloides [15]. Sua escolha para o contexto deste trabalho foi consequência da sua disseminação pela comunidade *creative commons* desde 2012 [16], resultando num grande número de recursos de áudio para uso pela comunidade científica.



Figura 1: Hatsune Miku (Crypton Future Media)

Considerando a tendência de crescimento da popularidade de vocaloides, segue a pergunta de pesquisa considerada: é possível utilizar processamento de sinais de áudio e aprendizado de máquina para identificar áudios destes vocaloides?

A extração de informações das músicas deste novo gênero pode contribuir para a pesquisa da evolução histórica da cultura (e.g., quais são os aspectos que levam à maior popularidade de determinada vocaloid?), auxiliar processos de análise de infração de direitos autorais (e.g., uso de vocaloide para ganhos comerciais sem a devida referência), e até fomentar discussões filosóficas e sociais sobre o tema (e.g., qual o impacto das vocaloides sobre as *idols* e outros cantores cujas músicas podem ser similares? As vocaloides contribuem para a padronização de músicas e podem limitar a liberdade artística de seus criadores?).

Este trabalho apresenta um experimento de processamento de áudio vocaloid realizando um estudo de caso da Hatsune Miku. Amostras de áudio humano e vocaloid foram obtidas do domínio público e utilizadas para treinar um classificador *Support Vector Machine* (SVM).

2. Trabalhos Relacionados

Uma busca nas grandes bases de pesquisa foi realizada com o intuito de identificar trabalhos relacionados com o aqui proposto. A *query* de busca utilizada foi "*vocaloid AND audio AND processing*", sem uma delimitação do período de publicação. A Tabela 1 apresenta a quantidade de artigos obtidos por base de pesquisa.

Tabela 1: Quantidade de artigos obtidos por base de pesquisa

Base de Pesquisa	nº de artigos
ACM	12
Elsevier	2
IEEE	1
Springer	17
Total	33

Devido ao baixo número de artigos obtidos, não foi necessário realizar um refinamento na *query* de busca. Após a leitura destes artigos em questão, inicialmente observando o resumo, e caso necessário lendo o corpo do texto, identificou-se que a maior parte deles propõe sistemas/soluções com funcionamento análogo ao de um vocaloid, ou que utilizem seus princípios e fundamentos, sem a realização de um estudo de caso envolvendo seu processamento de áudio, como é proposto neste artigo, ou então apenas uma menção à sua existência. Desta forma, este trabalho apresenta caráter de ineditismo, conforme suportado pelos resultados obtidos nesta pesquisa na literatura.

3. Método

O método aplicado, ilustrado pela Figura 2, consiste nas seguintes etapas:

1. **Coleta de Dados:** Obtenção de músicas sob licença *Creative Commons* instrumentais e de vocaloides. Neste passo destaca-se a dificuldade de obter músicas de cantores humanos de forma gratuita ou de forma ágil. O total de

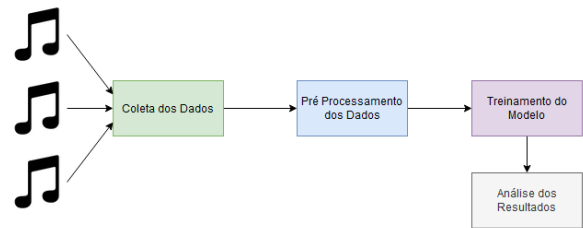


Figura 2: Método aplicado

arquivos de humanos também é mais limitado quando comparado aos arquivos disponíveis das vocaloides. Os arquivos instrumentais foram obtidos em <https://www.bensound.com/royalty-free-music/>, enquanto os arquivos das vocaloides foram obtidos em <http://mikudb.moe/>.

2. **Pré Processamento dos Dados:** Conversão da música em formato *mp3* para o formato *wav* compatível com a biblioteca *pyAudioAnalysis* utilizada. Foi utilizado o *FFmpeg* [17] por meio de chamada no notebook Python no ambiente de execução local Jupyter Notebook. Houve também o processamento dos áudios usando a biblioteca Python *pyAudioAnalysis* [18] para obter os parâmetros chroma. Por simplicidade, foram considerados apenas os 39 primeiros parâmetros de cada um dos 12 parâmetros chroma nos experimentos realizados.
3. **Treinamento dos Modelos:** Separação em bases de treinamento e teste, com 70% da base para treinamento e 30% para teste. Em seguida, seleção dos parâmetros chroma usando o método *Lasso* da biblioteca *sklearn* [19], e também a seleção de hiperparâmetros para posterior treinamento do classificador SVM usando o método *GridSearchCV* da mesma biblioteca *sklearn*. Por fim, treinamento do classificador SVM usando os hiperparâmetros obtidos.
4. **Análise dos Resultados:** considerando cenários previstos nos experimentos, analisar os resultados obtidos com o classificador com métricas acurácia, precisão, *f1 score*, além da curva *Receiver Operating Characteristic* (ROC). O uso da curva ROC está relacionado com o cálculo da área embaixo da curva (AUC), que apresenta um valor entre 0 e 1. Quanto mais próximo o valor obtido está do valor máximo, melhor o desempenho do classificador.

É importante destacar que a escolha do *Support Vector Machine* (SVM) foi motivado pelo seu bom desempenho na identificação de áudio sintético [20], e sua maior simplicidade quando comparado a técnicas com arquiteturas de redes neurais. O *Lasso* é um modelo linear que estima coeficientes esparsos, e seu uso foi motivado pela possibilidade de reduzir o número de parâmetros sob os quais uma solução é dependente [21], o que pode ajudar a evitar o *overfitting* do classificador.

Dois experimentos foram realizados. O primeiro consistiu em treinar um classificador para diferenciar músicas instrumentais de gêneros diversos das músicas da vocaloid Hatsune Miku. O segundo consistiu em treinar um classificador para diferenciar músicas de outras vocaloides (e.g., Kagamine Rin) das músicas da vocaloid Hatsune Miku.

No total foram utilizadas 60 músicas, divididas em três classes. A classe 1 possui 20 músicas de autoria da vocaloid Hatsune Miku, enquanto a classe 0 possui 20 músicas instrumentais de diversos gêneros. Por fim, a classe 0' possui 20 músicas de outras vocaloides. Desta forma, as músicas correspondentes as classes 1 e 0 foram utilizadas no primeiro experimento, enquanto as músicas correspondentes as classes 1 e 0' foram usadas no segundo experimento.

4. Resultados

Ambos os experimentos foram realizados utilizando o ambiente de execução local do *Jupyter Notebook*, em um computador com 8GB de RAM, processador Intel Core i3 em sistema operacional Windows 10.

A Figura 3 apresenta um chromograma de uma das músicas de Hatsune Miku obtido após processamento do áudio por meio da biblioteca *pyAudioAnalysis*. O chromograma é uma representação visual dos parâmetros chroma, apresentando graficamente no intervalo de tempo do áudio a intensidade de cada um dos 12 semitons de uma oitava {C, C#, D, ..., B}.

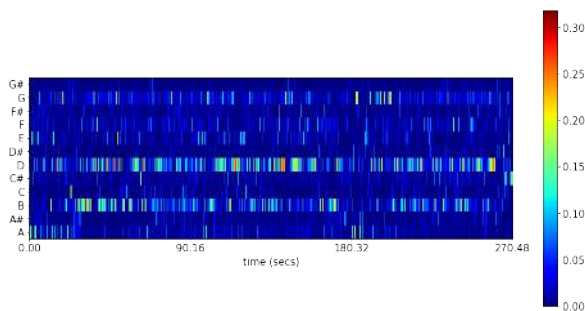


Figura 3: Chromograma de uma das músicas da vocaloid Hatsune Miku

O primeiro experimento é ilustrado na Figura 4. Arquivos de músicas instrumentais de gêneros diversos e músicas da vocaloid Hatsune Miku são processadas pela biblioteca *pyAudioAnalysis*. A partir do total de 468 parâmetros chroma, o método *Lasso* seleciona 18 parâmetros. Os hiperparâmetros obtidos com o *GridSearchCV* usando como métrica o *f1 score* e validação cruzada *5-fold* são $C = 5$, $\gamma = scale$ e $kernel = rbf$.

Foram coletadas métricas e traçada a curva ROC correspondente ao desempenho do classificador. Obteve-se uma acurácia de 81,67%, precisão de 80%, *recall* de 67%, e o *f1 score* de 62%. A Figura 5 apresenta a curva ROC, que mostra que o valor da área sob a curva foi de 0.83.

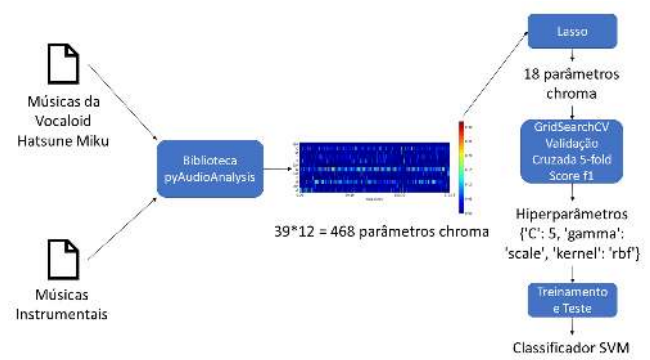


Figura 4: Experimento 1 de detecção de música da Hatsune Miku em comparação com músicas de outros gêneros

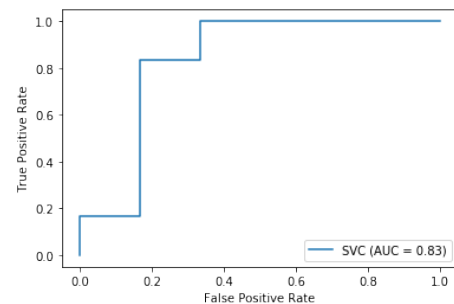


Figura 5: Curva ROC do Experimento 1

O segundo experimento é ilustrado na Figura 6. Arquivos de músicas de outros vocaloides e músicas da vocaloid Hatsune Miku são processadas pela biblioteca *pyAudioAnalysis*. A partir do total de 468 parâmetros chroma, o método *Lasso* seleciona 14 parâmetros. Os hiperparâmetros obtidos com o *GridSearchCV* são $C = 50$ e $kernel = linear$.

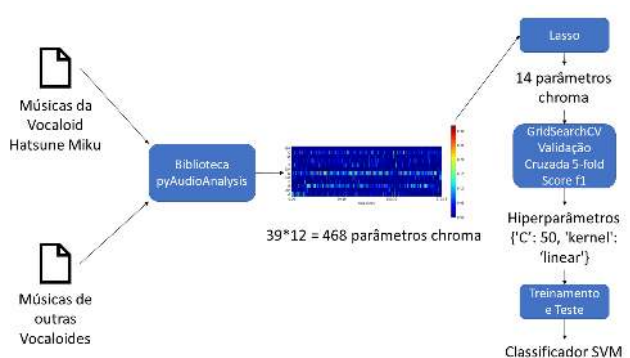


Figura 6: Experimento 2 de detecção de música da Hatsune Miku em comparação com músicas de outras vocaloides

Novamente foram coletadas métricas e traçada a curva ROC correspondente ao desempenho do classificador. Obteve-se uma acurácia de 80%, precisão de 88%, *recall* de 83%, e o *f1 score* de 83%. A Figura 7 apresenta a curva ROC, que mostra que o valor da área sob a curva foi de 0.86.

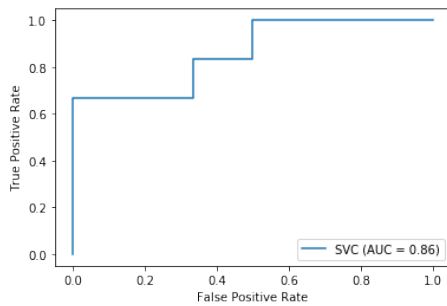


Figura 7: Curva ROC do Experimento 2

5. Conclusões

Este trabalho provou a viabilidade de obter um classificador que diferencia áudios da vocaloid Hatsune Miku de músicas instrumentais e de músicas de outras vocaloides, constituindo uma iniciativa na área de extração de informação de músicas deste novo gênero.

Como trabalhos futuros, destaca-se os testes e comparação com outros modelos de aprendizado de máquina, escolha de outros parâmetros dos áudios como o *Mel-Frequency Cepstral Coefficients* (MFCC), ou seleção de parâmetros a partir de técnicas de *deep learning* em conjunto com classificadores SVM [22]. Também é possível realizar testes com um conjunto maior de amostras da música, de forma a investigar o impacto no desempenho dos classificadores. Outra possibilidade é integrar vocaloides a um método para construção automática de bases de dados abertos de músicas para fomentar mais pesquisas na área.

Agradecimentos

Os autores agradecem a *Crypton Future Media* desenvolvedora da Hatsune Miku e toda a comunidade responsável por criar conteúdo e disponibilizá-lo sob licença *creative commons* viabilizando este trabalho.

Referências

- [1] Vladimir Kulyukin, Sarbajit Mukherjee, and Prakhar Amalthe. Toward audio beehive monitoring: Deep learning vs. standard machine learning in classifying beehive audio samples. *Applied Sciences*, 8(9):1573, 2018.
- [2] Hareesh Bahuleyan. Music genre classification using machine learning techniques. *arXiv preprint arXiv:1804.01149*, 2018.
- [3] Gerson Albuquerque Silva. Proposta de construção de um banco de dados de amostras de fala para uso forense em um arcabouço bayesiano. *Revista Brasileira de Criminalística*, 5(1):35–45, 2016.
- [4] Garima Sharma, Kartikeyan Umapathy, and Sridhar Krishnan. Trends in audio signal feature extraction methods. *Applied Acoustics*, 158:107020, 2020.
- [5] Roberto Font, Juan M Espín, and María José Cano. Experimental analysis of features for replay attack detection-results on the asvspoof 2017 challenge. In *Interspeech*, pages 7–11, 2017.

- [6] Xavier Serra. Creating research corpora for the computational study of music: the case of the compmusic project. In *Audio engineering society conference: 53rd international conference: Semantic audio*. Audio Engineering Society, 2014.
- [7] Mohamed Sordo, Amin Chaachoo, and Xavier Serra. Creating corpora for computational research in arab-andalusian music. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology*, pages 1–3, 2014.
- [8] Ajay Srinivasamurthy, Gopala Krishna Koduri, Sankalp Gulati, Vignesh Ishwar, and Xavier Serra. Corpora for music information research in indian art music. In *Georgaki A, Kouroupetroglou G, eds. Proceedings of the 2014 International Computer Music Conference, ICMC/SMC; 2014 Sept 14-20; Athens, Greece.[Michigan]: Michigan Publishing; 2014*. Michigan Publishing, 2014.
- [9] Hideki Kenmochi and Hayato Ohshita. Vocaloid-commercial singing synthesizer based on sample concatenation. In *Eighth Annual Conference of the International Speech Communication Association*, 2007.
- [10] Zhicong Lu, Chenxinran Shen, Jiannan Li, Hong Shen, and Daniel Wigdor. More kawaii than a real-person live streamer: Understanding how the otaku community engages with and perceives virtual youtubers. CHI '21, New York, NY, USA, 2021. Association for Computing Machinery.
- [11] Xin Zhou. Virtual youtuber kizuna ai. *ies, Lund U*, page 205.
- [12] RAISING THEIR. The hatsune miku phenomenon: More than a virtual j-pop diva. *The Journal of Popular Culture*, 49(5), 2016.
- [13] Niconico. Niconico. <https://www.nicovideo.jp/>, 2021. Acesso: 05/06/2021.
- [14] Bilibili. Bilibili. <https://www.bilibili.com/>, 2021. Acesso: 05/06/2021.
- [15] Hideki Kenmochi. Vocaloid and hatsune miku phenomenon in japan. In *Interdisciplinary Workshop on Singing Voice*, 2010.
- [16] Chiaki. Hatsune miku joins the cc community. <https://creativecommons.org/2012/12/14/hatsune-miku-joins-the-cc-community/>, 2021. Acesso: 05/06/2021.
- [17] FFmpeg-developers. Ffmpeg. <https://www.ffmpeg.org/>, 2021. Acesso: 05/06/2021.
- [18] Theodoros Giannakopoulos. pyaudioanalysis: An open-source python library for audio signal analysis. *PloS one*, 10(12):e0144610, 2015.
- [19] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *the Journal of machine Learning research*, 12:2825–2830, 2011.
- [20] Pavel Korshunov and Sébastien Marcel. A cross-database study of voice presentation attack detection. In *Handbook of Biometric Anti-Spoofing*, pages 363–389. Springer, 2019.
- [21] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.
- [22] Ivan Himawan, Srikanth Madikeri, Petr Motlicek, Milos Cernak, Sridha Sridharan, and Clinton Fookes. Voice presentation attack detection using convolutional neural networks. In *Handbook of Biometric Anti-Spoofing*, pages 391–415. Springer, 2019.

CREATES - Convolutional Neural Network applied to Identification of Musical Elements in OMR

Jenaro Augusto Barbosa*, Edimilson Batista dos Santos

¹Universidade Federal de São João del-Rei – UFSJ

Departamento de Ciência da Computação – DCOMP

Campus Tancredo Neves (CTAN) – São João del-Rei, MG

jenaro@ufsj.edu.br, edimilson.santos@ufsj.edu.br

Abstract. *Optical music recognition (OMR) is an important tool to recognize a scanned page of music sheet automatically, which has been applied to preserving music scores. In this paper, we present a comparative study among Convolutional Neural Network (CNN) and state-of-the-art classifiers to classify musical symbols. The initial results show that the CNN is promising in this task according to provided information.*

1. Introduction

A significant amount of musical works produced in the past are still available only as original manuscripts or as photocopies on date. In several historical cities, it is possible to find important music collections dated from the beginning of 18th century which are in this state. Thus, some works, such as Copista system [1], have proposed to apply computer science techniques to musical scores recognition through a process named Optical Music Recognition (OMR).

Optical Music Recognition (OMR) is a Computer Science field applied to music that deals with problems like recognition of handwritten scores. The applications in OMR are similar to Optical Character Recognition (OCR) tools. However, it is not a straightforward extension from the OCR since the problems to be faced are substantially different.

The OMR is needed for the preservation of musical works which requires digitalization and should be transformed into a machine readable format. An OMR program should thus be able to recognize the musical content and make semantic analysis of each musical symbol of a musical work. Generally, such a task is challenging because it requires the integration of techniques from some quite different areas, i.e., computer vision, artificial intelligence, machine learning, and music theory. In spite of existing applications that converts handwriting scores into editable scores, most of these applications a) do not work with manuscript scores[2], b) are very expensive and c) are not open source. All these reasons encourage to build a brand new tool on the OMR. The development of an OMR system can be divided into some distinct parts, e.g.: the image acquisition, image preprocessing and digital image recovery, the recognition of musical symbols with computer vision and machine learning, the music notation reconstruction and the symbolic music output. In

this paper, we focus on the step of recognition of musical symbols through machine learning techniques. Specifically, we propose to apply classification algorithms to identify such symbols, with emphasis on Convolutional Neural Networks, since it can obtain good results without many image processing steps.

The **CREATES** - Convolutional neuRal nEtwork Applied To idEntification muSical - solution can be divided into some stages. The separation occurs in four parts: (A) — raw state, (B) — preprocessing, (C) — CNN, (D) — Output, being a base solution for a future framework to be later implemented.

2. Related Work

Many musical works produced in the past are still currently available only as original manuscripts or as photocopies. The OMR is needed for the preservation of these works which requires digitalization and should be transformed into a machine readable format. Despite the many research activities on optical music recognition (OMR), the results for handwritten musical scores are far from ideal.

A lot of work on the OMR include staff lines detection and removal [3, 4, 5, 6], music symbol segmentation [7, 8], a tool proposal to convert handwriting scores into a digital music representation [1].

For many works, the machine learning area offers several methods to classify the music symbols. In [9], the authors propose a new combined neural network classifier, which has the potential to achieve a better recognition accuracy. In [10], a comparative study of several recognition algorithms of music symbols is presented. Although all the approaches have been shown to be effective in specific environments, they all suffer from some limitations. Thus, in this paper, we propose to apply Convolutional Neural Networks (CNNs) (see 3.1) in the task of classifying musical symbols for OMR systems. CNNs have obtained good results in image analysis and it can do without many image processing steps.

3. CREATES - CNN proposed to Recognize Musical Elements

Fig. 1 briefly presents the stages of the proposed solution for the problem of recognition of musical elements. The first two stages in Fig. 1, (A) and (B), are applied to the other classifiers and to the CREATES, being a part of the solution.

*Supported by 004/2019/PROPE/UFSJ.

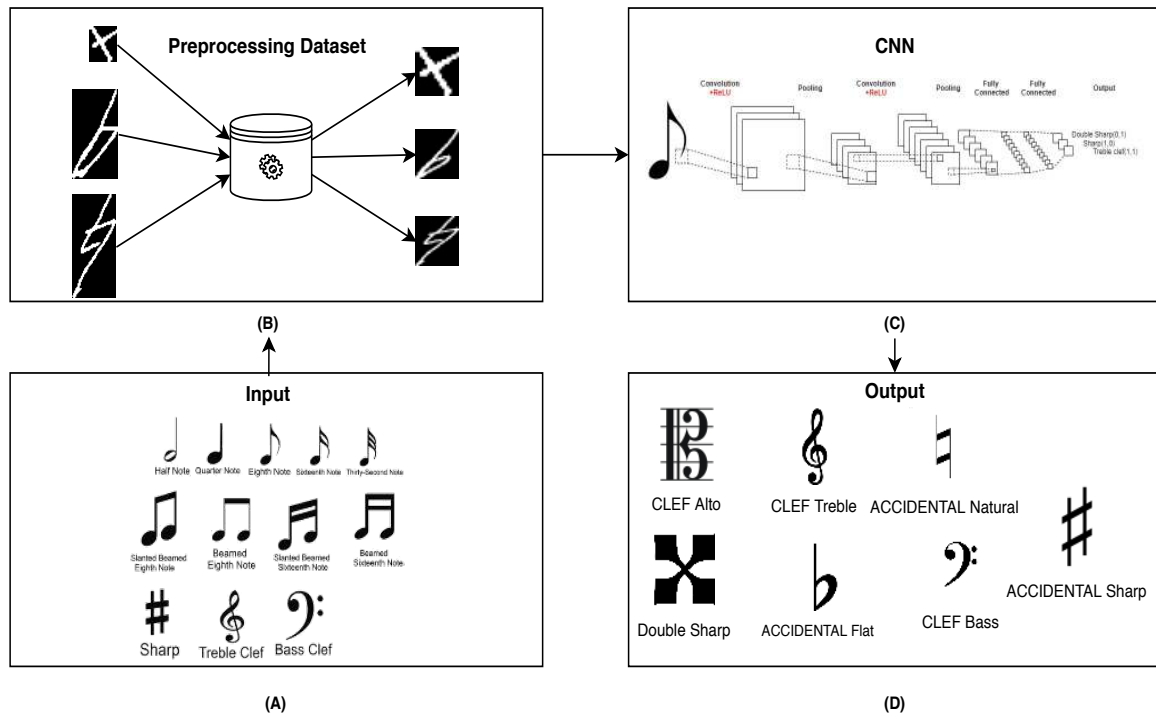


Figure 1: CREATES - Flowchart of the proposed solution.

The stage (A) - raw state - is the first stage with images in the initial state from the dataset [11]), which has 4094 instances of 7 types of musical symbols. These symbols can be seen in Figure 2. The images in the set have different dimensions and the set is not separated into a test and training set.



Figure 2: Each of the seven types of musical symbols used in the initial experiments.

The second stage is (B), where pre-processing in the images was performed to define a standard size (28x28). Here, the OpenCV library is used for this and also an interpolation process to resize. Next, a random separation process is carried out to select and generate training and test datasets. Thus, the original dataset was divided in 80% for training and 20% for testing. These pre-processed datasets were used in the experiments with the classifiers. Lastly, a numerical value was assigned to each musical symbol to label the classes (destination). Table 1 shows the number of instances for each musical symbol in the training and test datasets, after pre-processing .

The proposed solution, CREATES, comprises the use of the Convolutional Neural Network (CNN) for the classification of musical notes.

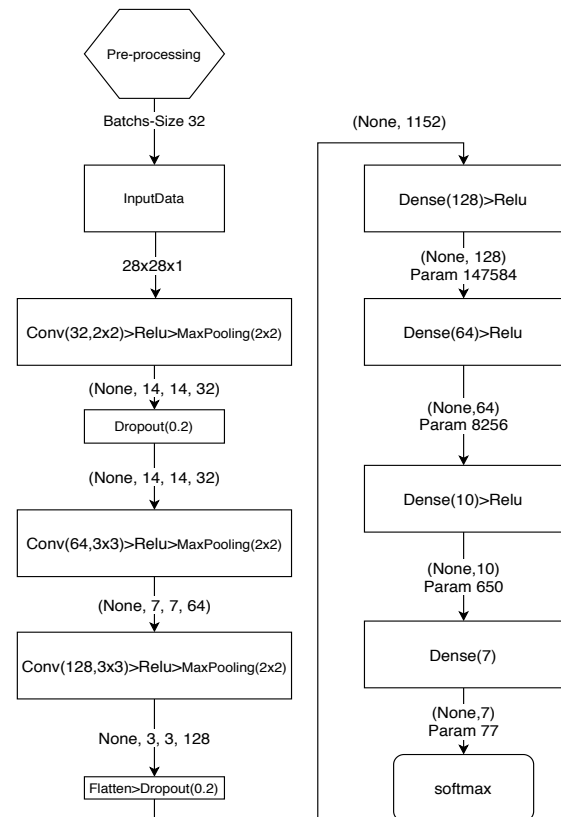


Figure 3: Proposed CNN architecture.

Table 1: Datasets to run the classification algorithms

Symbols	Instances ¹	Train	Test	Class
Alto	759	607	152	0
Double Sharp	497	397	100	1
Treble	820	656	164	2
Flat	517	413	104	3
Natural	471	376	95	4
Bass	549	439	110	5
Sharp	481	384	97	6
Total	4094	3272	822	

¹ Instance: number of instances for each symbol.

3.1. Proposed CNN architecture

In the third stage of the solution presented in Fig.1 (C), a CNN architecture was developed from the open source library known as TensorFlow to classify a set of musical symbols. The CNN's training and test have been performed on Google Colaboratory. For the training, the images have been divided in batch with size of 32 and use the Keras .

The CNN architecture is presented in Fig. 3. It can be divided in two parts: i) features extraction and ii) set of dense layers. In the features extraction step, the CNN architecture has 3 layers of convolution, 3 layers of pooling, and 1 regularization layer. The parameters, such as number of filters and kernel size, have been defined empirically. In all of these layers, the ReLu activation function is used, since it is popular in works of the area [7]. The regularization layer (dropout(0.2)) tries to avoid the overfitting, eliminating 20% of the output units from the previous pooling layer.

In the second part of the CNN architecture, we have the dense layers in addition to 1 transformation layer and 1 regularization layer. The first layer in the second part is of the Flatten type, which can be understood as a separation between the two parts of the network. Usually, this layer operates a transformation on the image matrix, changing it to an array. The regularization layer also applies a 20% discard rate on the value of the previous layer's output. All the dense layers use the ReLu activation function, except the last that uses the Softmax activation function. This CNN model has been compiled using the Adam optimization function, which is a stochastic gradient descent method, based on the adaptive estimation of first and second order moments.

4. Initial Experiments and Analysis of Results

In the initial experiments, we used CRATES in addition to the following state-of-the-art classifiers: SVM, KNN, Neural Network MLP, Gaussian Naive Bayes, Multinomial Naive Bayes, Complement Naive Bayes and Bernoulli Naive Bayes. For these classifiers, we use the Scikit-learn library and the same platform Google Colab, using the

training and test datasets presented in the first and second stages of the solution Figure 1.

The parameters of some algorithms were empirically defined using the randomized searchCV function available in the Scikit-learn library. Thus, the KNN has obtained better results with the number of neighbors $K = 2$. A multi-layer neural network was used, containing 5 hidden layers and 100 neurons in each. For the other algorithms applied in the experiments, the parameters remained with the default values defined in the library.

Table 2 shows the results of the classifiers when trained with grayscale images, using comparison metrics such as accuracy, precision, recall and F1-Score. CNN presented the best results (**92 %** of accuracy, **93 %** of precision, **92 %** of recall and **92 %** of F1-Score), presenting greater relation to the others classifiers, but with a longer time than the other classifiers. In this experiment, the SVM algorithm showed a second longer time compared to others classifiers, but it obtained low accuracy (**20%**), low recall (**14%**) and low F1-Score (**5%**) when compared to other state-of-the-art classifiers. We believe that the SVM algorithm was not able to learn the using image dataset in grayscale.

Table 2: Results (in %) obtained by classification algorithms using images

Classifiers	Acc	Pre	Recall	F1	Time
SVM	20	89	14	5	28.09s
KNN	64	65	65	64	4.86s
MLP	66	66	65	66	6.85s
Gaussian NB	60	60	62	60	0.068s
Multinomial NB	63	64	63	63	0.083s
Complement NB	55	59	59	58	0.078s
Bernoulli NB	63	64	65	64	0.097s
CNN	92	93	92	92	120s

As the accuracy may not be so reliable, since the classes do not have the same number of instances, Fig. 4 presents the confusion matrix obtained by proposed CNN. It is possible to note that the CNN maintains a good hit rate for each class, only with more errors in the flat class.

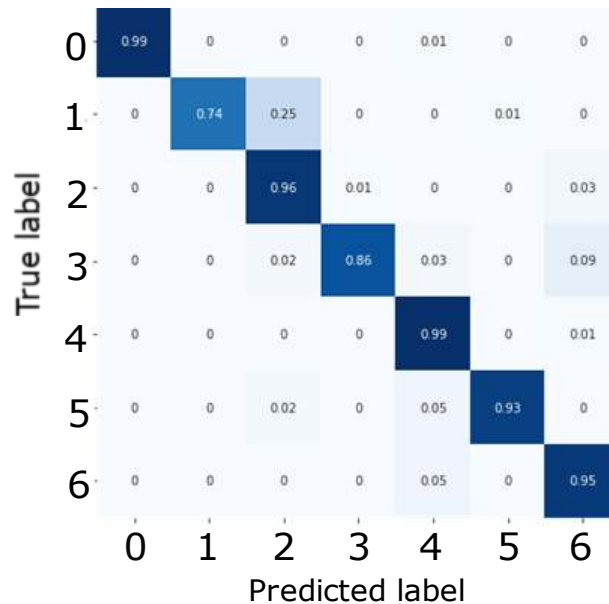


Figure 4: Confusion matrix obtained by proposed CNN architecture. Label Confusion Matrix: 0-Double Sharp, 1-Flat, 2-Natural, 3-Sharp, 4-Alto, 5-Bass, 6-Treble.

5. Conclusions

In this paper, it is proposed the investigation of algorithms and classification methods that can help in the recognition of musical elements for OMR applications. The initial proposal is to apply Convolutional Neural Networks, which is able to obtain good results in image analysis without performing much preprocessing.

The initial experiments provided results for the comparison of different classifiers. The proposed CNN proved to be adequate for the task of recognizing musical symbols.

According to the literature, it is noticed that the area of OMR (Character Recognition) manuscripts is still new and full of obstacles. The results obtained in this paper indicate that there is much to be done in this area. Thus, it is intended to continue the development of this project, taking advantage of the experience and the results acquired and applying to the development of a system to convert handwriting scores into a digital music representation.

References

- [1] Avner Maximiliano de Paulo, Flávio Luiz Schiavoni, Marcos Antônio de Matos Laia, and Daniel Luiz Alves Madeira. Copista-sistema de omr para a recuperação de acervo histórico musical. *XV SBCM-Computer Music: Beyond the frontiers of signal processing and computational models*, 2015.
- [2] David Bainbridge and Tim Bell. The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95–121, 2001.
- [3] Ana Rebelo and Jaime S Cardoso. Staff line detection and removal in the grayscale domain. In *2013 12th International Conference on Document Analysis and Recognition*, pages 57–61. IEEE, 2013.
- [4] Christoph Dalitz, Michael Droettboom, Bastian Pranzas, and Ichiro Fujinaga. A comparative study of staff removal algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5):753–766, 2008.
- [5] Ichiro Fujinaga. Staff detection and removal. In *Visual Perception of Music Notation: On-Line and Off Line Recognition*, pages 1–39. IGI Global, 2004.
- [6] Jaime dos Santos Cardoso, Artur Capela, Ana Rebelo, Carlos Guedes, and Joaquim Pinto da Costa. Staff detection with stable paths. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(6):1134–1139, 2009.
- [7] Florence Rossant and Isabelle Bloch. Robust and adaptive omr system including fuzzy modeling, fusion of musical rules, and possible error detection. *EURASIP Journal on Advances in Signal Processing*, 2007(1):081541, 2006.
- [8] Alicia Fornés, Josep Lladós, and Gemma Sánchez. Primitive segmentation in old handwritten music scores. In *International Workshop on Graphics Recognition*, pages 279–290. Springer, 2005.
- [9] Cuihong Wen, Ana Rebelo, Jing Zhang, and Jaime Cardoso. A new optical music recognition system based on combined neural network. *Pattern Recognition Letters*, 58:1–7, 2015.
- [10] Ana Rebelo, G Capela, and Jaime S Cardoso. Optical recognition of music symbols. *International Journal on Document Analysis and Recognition (IJDAR)*, 13(1):19–31, 2010.
- [11] Alicia Fornés, Josep Lladós, and Gemma Sánchez. Old handwritten musical symbol classification by a dynamic time warping based method. In *International Workshop on Graphics Recognition*, pages 51–60. Springer, 2007.

Modelo automático de classificação de gêneros musicais amazônicos

Douglas Silva¹, Lucas Zampar¹, Felipe Rodrigues¹, Cláudio Gomes¹

¹Departamento de Ciências Exatas e Tecnologias - Universidade Federal do Amapá (UNIFAP)
Caixa Postal 261 – 68.906-970 – Macapá – AP – Brazil

{douglasdasilvasousa8, lucas.26.zampar, leepe.fr}@gmail.com, claudiorogério@unifap.br

Resumo. A globalização afeta a preferência musical da sociedade atual, que apresenta continua estima por ou gêneros musicais internacionais em detrimento aos nacionais ou locais. A música é dos meios de comunicação utilizados para a construção e organização da estrutura social influenciando o estilo de vida, gostos e convivência interpessoal. Portanto, cada manifestação de gênero musical apresenta função distinta para um ouvinte, como por exemplo, para dançar, festejar, descansar, ajudar na solidão, na tristeza etc. Diversos aplicativos como spotify e soundcloud, utilizam-se de classificadores de gêneros musicais para indicar, prever ou sugerir novas músicas para seus ouvintes. Por diversos motivos, a maioria dos classificadores não possui informações de gêneros musicais regionais. Este trabalho propõe um modelo de classificação automático de gêneros musicais populares amazônicos. Inicialmente, confeccionou-se uma base de dados contendo os gêneros musicais: andino, brega, carimbó, cúmbia, marabaixo, pasillo, salsa e vaqueirada, oriundos da região da Amazônia Legal dos países: Brasil, Guiana Francesa, Venezuela, Colômbia, Equador, Bolívia e Peru. Para a construção da base de dados, extraiu-se diversas características de cada música ao total de 64 parâmetros. Analisou-se os modelos de aprendizado de máquina na qual XGB, KNN, SVM e MLP obtiveram acurácia de 67.62%, 74.12%, 71.35%, 76.13%, respectivamente.

Palavras-chave — aprendizado de máquina, recuperação da informação musical, sistemas de classificação

Abstract. Globalization affects the musical preference of today's society, which has continued esteemed by international rhythms compared to the national or local. Music is a form of communication used for the construction and organization of the social structure, which It influences lifestyles, tastes, and interpersonal relationships. Therefore, each musical genre manifestation has different functions for a listener, for example, to dance, celebrate, rest, help in solitude, sadness, etc. Several applications use music genre classifiers to indicate, preview, or suggest new music for their listeners. For various reasons, most classifiers do not have information on regional music genres. This work proposes an automatic classification model for Amazonian popular musical genres. Initially, a database was created containing the musical genres: andino, brega, carimbó, cúmbia, marabaixo, pasillo, salsa e vaqueirada, from the Legal Amazon region of the countries: Brazil, French Guiana, Venezuela, Colombia, Ecuador, Bolivia and Peru. For the construction of the database, several characteristics of each song were extracted from a total of 64 parameters. Machine learning models were analyzed in which XGB, KNN, SVM e MLP obtained an accuracy of 67.62%, 74.12%, 71.35%, 76.13%, respectively.

Keywords — machine learning, music information retrieval, clas-

sifiers

1 Introdução

A globalização afeta a preferência musical da sociedade que tem continuo estima por ritmos ou gêneros musicais internacionais em detrimento à nacionais ou locais. A música é uma forma de expressão invisível, não palpável e ainda maleável tornando-se difícil a tarefa de descrevê-la, rotulá-la ou classificá-la. A base de qualquer música é composta por o ritmo, melodia e harmonia. O ritmo é a pulsação contínua que manifesta a marcação temporal regular de uma música, como por exemplo bater palmas ou estalar os dedos [1]. Além do ritmo, há ainda a melodia que é uma sequência de notas tocadas por instrumentos. Por fim, o último elemento básico da música é a harmonia, responsável em acompanhar a melodia a partir de notas tocadas simultâneas e sequenciais (acordes). A harmonia dá sentimento para a música, é responsável em gerar músicas com aspectos alegres, melancólicos, tristes etc.

O autor Menezes [2] comenta que o Brasil consome em maior quantidade, música popular brasileira que a música internacional, indicando a possibilidade de “reinvenção brasileira” a cada momento. Menezes ainda afirma a relevância da contribuição da música popular brasileira para a compreensão sobre as músicas populares do mundo. Dessa forma, o estudo sobre músicas regionais populares da Amazônia tem grande valor para o cenário musical regional, nacional e internacional.

Com base em intenções, objetivos ou valores históricos e culturais, um gênero musical é caracterizado a partir de propriedades comuns ou similares relacionadas à instrumentação, estrutura rítmica e conteúdo harmônico extraídos a partir de várias músicas [1]. Portanto, cada manifestação rítmica apresenta funções distintas para o ouvinte, como por exemplo, para dançar, festejar, descansar, ajudar em momentos de solidão, na tristeza etc.

Atualmente, diversos aplicativos utilizam-se de classificadores para indicar, prever ou sugerir novas músicas para seus ouvintes. Basicamente, a cada nova interação do usuário ao aplicativo, os classificadores adquirem novos conhecimentos e aprendem novas opções musicais para os ouvintes. Assim, tanto o usuário quanto os classificadores são duplamente consumidores de um sistema com propósitos similares e complementares [1].

Para classificar gêneros musicais é necessário, inicialmente, confeccionar uma base de dados, que em grande parte é constituído de ação manual, para prosseguir através de diversos modelos computacionais de processamento e classificação. Na literatura há várias arquiteturas que realizam o processo automático de classificação de gêneros musicais, porém a maioria não possui informações de gêneros musicais regionais ou, por diversos motivos, estão apenas engajados na classificação de gêneros musicais internacionais [2]. Assim, ouvintes que buscam ritmos regionais ou locais podem ter grande desconforto de sugestões por limitações de classificadores [3].

Este trabalho propõem um modelo de classificação de gêneros musicais amazônicos. Inicialmente, criou-se uma base de dados a partir da plataforma de *streaming youtube* contendo os ritmos musicais da amazônicos envolvendo os países: Brasil, Guiana Francesa, Venezuela, Colômbia, Bolívia e Peru. Os principais ritmos e gêneros analisados foram: marabaixo, brega, vaqueirada, carimbó, cumbia, merengue, salsa, pasillo e andino.

O artigo está organizado na seguinte forma. A seção 2 apresenta discussões de trabalhos relacionados. Na seção 3, apresenta-se o classificador automático de ritmos amazônicos e os resultados obtidos e, por fim, a seção 4, comenta-se as conclusões e trabalhos futuros.

2 Trabalhos Relacionados

O trabalho de Guggari et al. [4] apresenta uma abordagem de classificação de ritmos a partir de árvore de decisão. Realizou comparativos entre os algoritmos de floresta aleatória, C4.5, bagging e adaboost em 15 base de dados. A quantidade de classes e parâmetros utilizados para a extração de características musicais foi dependente de cada base de dados. Comparando-se as técnicas de classificação com as bases de dados, obteve-se diferentes resultados, em que no pior ao melhor caso alcançou-se 50% a 100% de acurácia. Porém, não pode-se afirmar qual a técnica mais indicada para a criação de um classificador de ritmos, pois cada base de dados possuiam quantidade de amostras, parâmetros e classes de ritmos diferentes.

O trabalho de Karunakaran e Arya [5] apresenta um classificador baseado nas técnicas de K-vizinhos mais próximos, máquina de vetor suporte, *naive Bayes*, redes neurais, análise do discriminante quadrático e lógica *fuzzy*. Investigou as classes de pop, rock e músicas eletrônicas nas bases de dados GTZAN [6] e *Free Music Archive* [7] com acurácia de 90% e 70% respectivamente.

O trabalho de Silla Jr. et al. [8] realizou um classificador automático de músicas latina. Para isso, confeccionou a base de dados intitulada *Latin Music Database (LMD)* [9] que possui 3160 músicas de 10 gêneros musicais: tango, bolero, bachata, salsa, merengue, axé, forró, sertaneja, gaúcha e pagode. Esta base possui três opções: utilização de 30 segundos iniciais, 30 segundos intermediário e 30 segundos finais de cada música. O classificador apresentou melhor resultado com características utilizando o segmento intermediário de cada da música. Realizou vários comparativos entre os modelos de redes neurais, máquina de vetor suporte e *naive bayes* na qual dependendo do cenário obteve em seu melhor caso a acurácia de 65,73%.

3 Classificador de ritmos populares da amazônia

Inicialmente, para classificar gêneros musicais é necessário, confeccionar uma base de dados, que em grande parte é constituído de ação manual, para em seguida, prosseguir a etapa de classificação por diversos modelos computacionais.

Na literatura há vários modelos que realizam o processo automático de classificação de gêneros musicais, porém a maioria não possui informações de gêneros musicais populares da amazônia ou, por diversos motivos, estão apenas engajados na classificação de gêneros musicais internacionais. Assim, ouvintes que buscam ritmos regionais ou locais podem ter grande desconforto de sugestões por limitações de classificadores.

A subseção 3.1 descreve características dos gêneros musicais escolhidos. A subseção 3.2 apresenta as características

utilizadas para a construção da base de dados dos gêneros musicais populares da amazônia. E, por fim, a subseção 3.3 apresenta as abordagens utilizadas para o modelo de classificador de gêneros musicais da amazônia.

3.1 Seleção dos gêneros musicais populares da Amazônia

Dessa forma, a partir da plataforma de *streaming youtube* realizou a seleção de, em média, 100 músicas para cada ritmo de: andino, brega, carimbo, cumbia, marabaixo, merengue, pasillo, salsa e vaqueirada. As músicas andinas são gêneros populares na Cordilheira dos Andes que inclui os países da Bolívia, Equador e Peru. Para cada gênero, utilizou-se mais de 6 grupos musicais diferentes.

Trazido de zona rural da Colômbia, a cumbia possui várias vertentes regionais em todo o país com características de folclore e celebrações festivas. Nos países da Colômbia, Panamá, Guianas, República dominicana, Panamá e região norte do Brasil o merengue apresenta conotação de celebração festiva bastante popular. O pasillo é outro ritmo de origem colombiana com grandes variações encontradas nas, principalmente, nos países da Colômbia, Panamá e Equador. A salsa é um dos ritmos mais bailados do mundo com várias influências rítmicas com rica sonoridade nos países da Colômbia, Porto Rico, México, Venezuela e Cuba.

O gênero do brega é um ritmo característico da região norte do Brasil, especificamente do estado do Pará, com influências de vários gêneros musicais com características dançantes oriundos da América Central, reconhecido como patrimônio cultural do Brasil. Outro ritmo da região norte do Brasil é o carimbó que apresenta características indígenas e influências das cumbias de folclore, é reconhecido como patrimônio cultural do Brasil. O marabaixo, reconhecido como patrimônio cultural do Brasil, é um ritmo popular da região norte do Brasil, especificamente no estado do Amapá, sobre influências rítmicas africanas e indígenas. A vaqueirada, ritmo popular do estado de Amazonas no norte do Brasil, apresenta características de ritmos de ciranda, toada e com encenação popular com diversas influências indígenas, africanas também reconhecido como patrimônio cultural do Brasil.

3.2 Criação da base de dados

A partir dos gêneros selecionados, comentados na subseção 3.1, extraiu-se características do tempograma, *chroma short-time Fourier transform* (*chroma.stft*), *chroma Constant Q spectral Transform* (*chroma.cqt*), RMS e *zero crossing rate* (ZCR) ao total de 64 parâmetros. Alguns desses parâmetros citados possuem diversos dados de um vetor ou matriz, dessa forma realizou-se testes estatísticos de média, moda e mediana criando 3 bases de dados, uma para cada teste estatístico. Ao final, a mediana apresentou melhor acurácia comparada com a média e moda.

O tempograma indica uma matriz de prevalência de certo tempo para cada momento temporal. Os testes mostraram que o uso do tempograma tiveram melhor acurácia caso fossem utilizados isoladamente. Portanto, o tempograma auxilia na identificação da velocidade dos ritmos. O *chroma.stft* apresenta características relacionadas às oitavas musicais, assim, auxilia na caracterização de tonalidades musicais. O *chroma.cqt* realiza de modo semelhante ao *chroma.stft* porém em escala logarítmica com características estatísticas para cada amostra no tempo, considerando questões de articulação, arpeggio e acordes musicais. Apesar de apresentarem características conceituais próximas, os

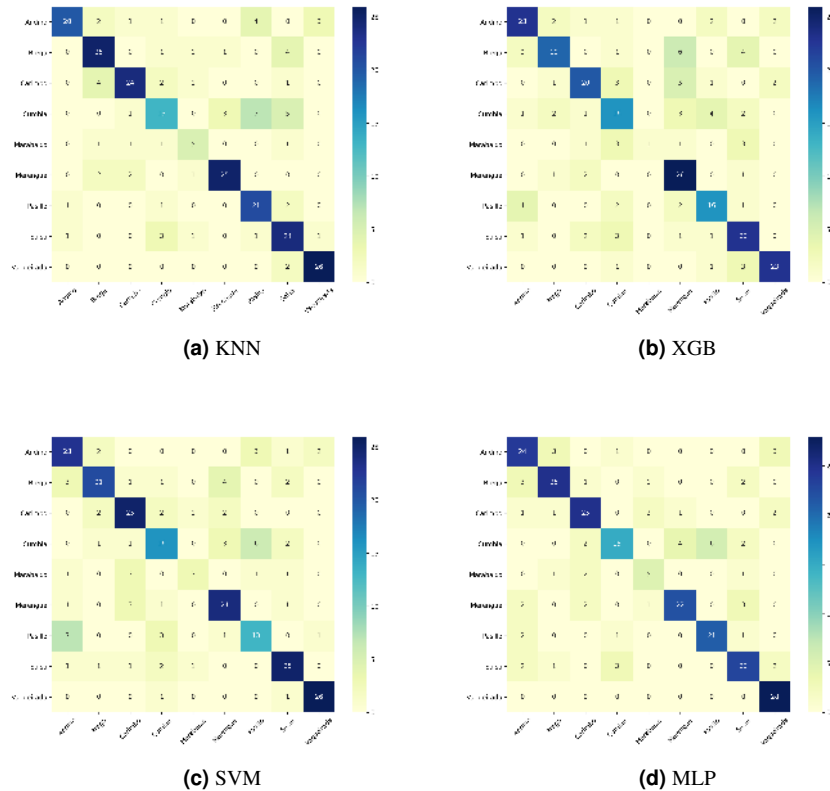


Figura 1: Matriz de confusão de testes entre modelos computacionais

Parâmetro	Índices das matrizes
Tempograma	[1 - 45]
Chroma stft	[1 - 5]
Chroma cqt	[1 - 12]
RMS	[1]
ZCR	[1]

Tabela 1: Índices utilizados da matriz de cada parâmetro

resultados alcançados em conjunto foram melhores caso utilizasse separadamente. Utilizou-se o *root-mean-square* (rms) devido a grande maioria das músicas não estarem equalizadas e por fim o *zero crossing rate* realiza avaliação de mudanças rápidas do sinal pelo cruzamento por zero. A tabela 1 apresenta os índices utilizados para cada parâmetro.

3.3 Classificador de gêneros musicais amazônicos

Conforme as considerações descritas na subseção 3.2, gerou-se a base de gêneros musicais populares amazônicos¹. Em seguida, realizou-se um estudo dos parâmetros relevantes para o classificador, conforme a subseção 3.2. Em sequência, realizou-se as análises de modelos computacionais entre *KNN*, *gradiente boosting*, *multilayer perceptron* e máquina de vetor suporte (SVM). Para todos os modelos, utilizou-se 70% da base de dados como treino e 30% como testes.

¹<https://bit.ly/3keCFuK>

No modelo *KNN* a partir de testes iniciais, obteve-se melhores resultados com os hiperparâmetros de 10 *k-vizinhos* e com análise de parâmetros considerando *distância*. O *gradiente boosting* (XGB) é um modelo computacional que realiza pequenas previsões para um conjunto de dados corrigidos temporalmente o preditor final. Assim, o XGB analisa a contribuição de cada preditor sequencial através de seu crescimento baseado no erro residual do preditor anterior. Com relação modelo de máquina vetor suporte (SVM), tenta encontrar a distância de linhas para a separação entre os dados das classes. O SVM é indicado para um volume de dados não muito grande, que não estejam sobrepostos, ou ainda com pouca quantidade de ruídos. Para o SVM, utilizou-se a função de decisão *ovr*, *kernel rbf* com taxa de 0.00205 e com penalidade *L2*.

Com relação às redes neurais *perceptron* (MLP), é outro modelo computacional de decisão com objetivos de classificar, agrupar e ainda prever informações a partir de dados repassados a rede de neurônios que com suas interconexões realizam o aprendizado do dados. É um modelo bastante complexo e inerente de sua estrutura possuindo diversas variáveis controláveis para facilitar resultados esperados [10]. Para o classificador, utilizou-se uma rede com 1520, 300, 100 por 9 camadas escondidas utilizando *tangente hiperbólica*, *relu*, *tangente hiperbólica* e *softmax* com 100 épocas e *batch size* de tamanho 32.

A figura 1 apresenta os resultados de cada modelo computacional. O XGB, KNN, SVM e MLP obtiveram acurácia de 67.62%, 74.12%, 71.35%, 76.13%, respectivamente. Em todos os modelos, percebe-se a dificuldade em classificar gêneros musicais do marabaixo, pois teve-se apenas o total de 30 músicas. Apresentou-se dificuldade em encontrar músicas do marabaixo. Com relação à cumbia e merengue, percebe-se algumas falhas na

classificação, pois esses gêneros influenciam os demais gêneros selecionados.

Dessa forma, os modelos KNN e SVM que possuem por essência realizar separação de classes apresentaram resultados semelhantes com relação ao resultado de acurácia quanto aos erros de classificação. Já os modelos XGB e MLP com características de aprendizado temporal, apresentaram evoluções de classificação dependendo de seus respectivos hiperparâmetros. No geral, analisando os erros dos classificadores pode-se considerar aceitáveis pois, os ritmos que foi classificados possuem sonoridade próxima umas das outras como a brega, carimbó, cúmbia, merengue e pasillo.

4 Conclusões e trabalhos futuros

Conclui-se que o modelo de redes neurais apresentou melhores resultados com relação aos demais modelos. Para trabalhos futuros, pretende-se analisar outros modelos de aprendizado de máquina com objetivo de alcançar melhorias na acurácia da classificação e ainda desenvolver melhor compreensão entre os ritmos populares da amazônia selecionados. Além disso, adicionar novas músicas para cada gênero musical afim de melhor compreender os ritmo por diversas características computacionais.

Referências

- [1] Susan Schreibman, Ray Siemens, and John Unsworth. *A new companion to digital humanities*. John Wiley & Sons, 2015.
- [2] Rafael José de Menezes Bastos. *As contribuições da música popular brasileira às músicas populares do mundo: diálogos transatlânticos Brasil-Europa-África - Segunda parte*. Universidade Federal de Santa Catarina, 2007.
- [3] Andre Holzapfel, Bob Sturm, and Mark Coeckelbergh. Ethical dimensions of music information retrieval technology. *Transactions of the International Society for Music Information Retrieval*, 1(1):44–55, 2018.
- [4] Shankru Guggari, Vijayakumar Kadappa, V Umadevi, and Ajith Abraham. Music rhythm tree based partitioning approach to decision tree classifier. *Journal of King Saud University-Computer and Information Sciences*, 2020.
- [5] Nagamanoj Karunakaran and Arti Arya. A scalable hybrid classifier for music genre classification using machine learning concepts and spark. In *2018 International Conference on Intelligent Autonomous Systems (ICoIAS)*, pages 128–135. IEEE, 2018.
- [6] Bob L. Sturm. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *CoRR*, abs/1306.1461, 2013.
- [7] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.
- [8] Carlos N Silla Jr, Celso AA Kaestner, and Alessandro L Koerich. Automatic music genre classification using ensemble of classifiers. In *2007 IEEE International Conference on Systems, Man and Cybernetics*, pages 1687–1692. IEEE, 2007.
- [9] Carlos Nascimento Silla Jr, Alessandro L Koerich, and Celso AA Kaestner. The latin music database. In *ISMIR*, pages 451–456, 2008.
- [10] Aurélien Géron. *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn & TensorFlow*. Alta Books, 2019.

Camomile-ELSE: creating plugins with Pure Data patches and the ELSE library

Astorga, E.M.V., Porres, Alexandre Torres

Núcleo de Pesquisas em Sonologia – Escola de Comunicação e Artes da Universidade de São Paulo
Av. Prof. Lúcio Martins Rodrigues, 443 – 055088-020, São Paulo, SP

EL Locus Solus (independent center) São Paulo, SP

esteban.astorga@usp.br, el.locus.solus@gmail.com

Abstract

Camomile is an open source project that turns patches made in the Pure Data programming language into plugins (VST, VST3, LV2 and Audio Unit) so you can load them in any plugin host on Windows, Linux and MacOS. We have a fork of this project that embeds the ELSE library of Pure Data externals into camomile so you don't have to only rely on the limited set of native Pure Data objects.

This paper describes the potential of the Camomile project, how it makes the creation of plugins very accessible to people without a background in computer science and also how this process is improved by including the extensive ELSE library. We also offer an overview of similar projects and the technical details of including the ELSE library into the camomile project.

1. Introduction

Pure data (a.k.a Pd) is an Open Source cross platform visual programming environment for interactive computer music and multimedia created by Miller Puckette in the mid 1990s. Pd is distributed for Linux, macOS and Windows. Pure Data has also been ported to iOS (Pd Party) [1] and Android (PdDroidParty) [2] operating systems thanks to libpd¹ [3].

Pure Data offers a very limited set of native objects, it misses lots of functionalities if compared to other similar environments such as Csound, SuperCollider and MAX. The Pd community has always relied on a large collection of external libraries and this also promoted forks of Pure Data that include pre-installed libraries. Nonetheless, you can currently easily install externals into the original Pure Data distribution (also known as 'Pd Vanilla').

On the other hand, Pure Data for iOS and android are only 'Vanilla' (meaning, no externals). It's not that you can't use externals with libpd, but most of the projects based on libpd are offered compiled only to include the native Pd objects. The Camomile project is yet another project that relies on libpd to embed Pure Data into an application. By default, it only allows you to

use Vanilla objects. Actually, not even that as it misses some objects like [pd~] and the graphical Data Structures objects.

We have forked Camomile to compile it with externals from the ELSE library [4], which is a library that's still in an early and experimental phase, but with a huge set of externals. ELSE has been in active development since the begging of 2017 and a first beta version was presented in SBCM that year [6]. At the time of this writing, ELSE is at beta version 42.

2. The breakthrough of Camomile (History)

Bridging Pd with other softwares is not hard via softwares like jack and MIDI bridges, but this can be clumsy and limiting when trying to bring Pure Data into the workflow other softwares that process audio. Hence, a proper integration like MAX has with MAX4LIVE has always been and attractive goal. One big advantage is to render or bounce the project inside your desired software with your Pd patches as part of the process.

Attempts in that direction start with Radium, which is a DAW² with a tracker like interface and has Pd embedded in its Linux version, allowing us to run pd patches since 2013. Later, in 2016, we first had the possibility of running pd patches in any DAW via VST support with PD Pulp [7], which is basically a VST plugin (built using JUCE [8] and libpd) that runs Pd patches. Its interface has 10 knobs that can be assigned to any parameter in the patch and buttons to manage the patch loading. PD Pulp's limitations are the lack of MIDI support, the impossibility to use more than one instance of the plugin and the static interface which doesn't allow any customization by the user. PD Pulp was abandoned after Camomile was released by Pierre Guillot in 2016 [9].

Camomile solves most of the problems PD Pulp had and offers more features. In the first public release³,

¹ LibPd turns Pd into an embeddable audio synthesis library that can use to create other applications

² Digital Audio Workstations, a kind of software that allows multi music production with track recording
³ See:
<https://web.archive.org/web/20210711174627/https://forum.pdpatchrepo.info/topic/9884/camomile-an-audio-plugin-that-loads-pure-data-patches>

Camomile could already deal with multiple instances of the plugin with a wrapper written by Pierre Guillot, because at that time LibPd had this limitation which was solved after this release⁴. Camomile also has MIDI support and provides the ability to create the plugin's interface. This opened the possibility for developers to design their own plugins and distribute them without the need of a Pd patch, and users can be totally unaware it was implemented with Pure Data.

Camomile is also built with LibPd and JUCE and can be used to compile and create VST3, AU and LV2 plugins in Linux, Windows and macOS operating systems. This covers virtually all the DAWs in use these days, plus other hosts like sound editors and other softwares.

Camomile allows the regular Pd user to create and distribute its own plugin without a background in computer science. However, the process of compiling isn't an obvious thing to all Pd users. The way Camomile compiles plugins is simple enough because knowledge about the JUCE environments isn't required and can be done with a minimum set of instructions, provided by tutorials available in the Wiki session of the official Camomile's repository. Compiling scripts are provided for Linux and MacOS, but Windows compilation, at the moment, needs this set of instructions (which is also valid for all operating systems).

In short, Camomile provides support for multiple instances, basic Pd GUI objects (toggle, slider, radio, comment, numbox and array graph), up to 64 automatable parameters with automatic recognition (name, label, range, minimum, maximum, etc.), MIDI in and out, play head position, BPM and up to 16 audio channels.

One current limitation is the lack of support for externals, forcing developers to work with the limited set of Vanilla objects. So we have developed a version of Camomile that includes objects from the ELSE library, which makes the process of creating plugins much easier and friendlier.

3. The ELSE library

The ELSE library [4] is a huge collection of externals and provides many ready made building blocks of computer and electronic music, such as filters, oscillators, noise generators, envelopes, sequencers, random generators, tools for algorithmic composition, etcetera...

While most of these could be implemented with Vanilla objects, the amount of work would be quite huge and you'd need a deep knowledge of Digital Signal

Processing and Pure Data. Not only that, but ELSE also provides things you cannot have with only Pd Vanilla.

ELSE makes the patching process much easier and accessible to the general public. Since Camomile's main selling point is that it allows people without a background in computer science to develop plugins, the ELSE library takes this premise to a whole new level, where you don't have to be a Pd expert and you can more intuitively program and create instruments, effects, MIDI processors and controllers.

ELSE has a structure that resembles the one of modular synthesizers and also has many ready made effects (chorus~, flanger~ and phaser~ for instance), plus advanced algorithmic composition tools like markov chain.

Originally, ELSE also comes with a vast computer music tutorial, with over 430 examples that covers a wide range of synthesis, algorithmic composition and DSP techniques. All of it just has a single dependency of ELSE externals, so you can easily create plugins with all that is provided in this tutorial.

4. Technical Details

In order to provide a version of Camomile-ELSE [5], Camomile with support for ELSE externals we had to recompile Camomile with the externals' code embedded into Camomile's source code. Camomile's main dependencies are JUCE and LibPd. JUCE provides classes to manage audio I/O, audio processing, media reading and writing and the hooks to compile plugins VST, LV2 (for Linux, Windows and MacOS) and AU plugins (MacOS only). All our compiling tests are done using a MacOS machine, a Virtual Machine with Ubuntu 12.04 LTS and another old notebook running Windows 10.

The process of setting Camomile to be compiled with ELSE externals is documented in Camomile repository⁵, it consists in adding the externals from ELSE into LibPd and then set JUCE to compile Camomile with these dependencies. Some modifications to the original code of ELSE were necessary in order to conform with limitations and conflicts with the Windows compiler.

The ELSE library consists in a collection of externals that are binaries (compiled from C code) and abstractions (Pd patches, where many use objects from the ELSE library). The binaries are compiled with LibPd and this procedure makes Pd see the ELSE externals from ELSE as if they were Pd native objects. This means you cannot load the externals with namespaces (as in `else/sine~`), but originally all the abstractions from ELSE that has externals from ELSE use namespaces in order to

⁴ See: <https://web.archive.org/web/20210711174255/https://github.com/libpd/libpd/issues/13>

⁵ See: <https://github.com/pierreguillot/Camomile/issues/214#issuecomment-704670696>

force loading the correct object. So we have a specially modified abstraction folder without namespaces and use the [declare] object (as in: [declare -path else]). This allows Camomile to find the abstractions inside the provided modified *else* folder, that you also need to include in your compilation project.

This allows support for all objects in the ELSE library (both binaries and abstractions) but most GUI objects aren't functional because of a limitation from Camomile. Objects from the ELSE library that are binaries needed to be ported like the native Pd GUIs were, and many GUI abstractions use data structures, which are also not implemented in Camomile. The discussion related can be found in Camomile repository⁶ and the difficulty consists in reimplementing Pd's Data Structure drawing features in the JUCE interface. So we are looking for other workarounds to provide better GUI support in Camomile.

Camomile-ELSE is compiled on the last stable version of Camomile, 1.0.7. There is already available a beta version 1.0.8 that supports the *pd~* object. This object makes it possible something like open Pd inside Pd, it was created to allow open and control different DSP chains in different threads controlled by operational system. On the last beta version of Camomile, the *pd~* object can be used to open the Pd application and the patch once you call the plugin. This can be great for Pd users but is not an ideal situation for the distribution of a plugin for non Pd users as it basically just opens Pd inside your plugin host and exposes the inner workings. On the other hand, this process allows you to run Pd with any externals that it can load, because now we're not compiling the externals in camomile anymore, just opening a Pd binary with its included external binaries. Nonetheless, this new camomile feature does a good job by not requiring the user to install Pd, the externals and the patch, because all of this is embedded in the generated plugin.

Besides the non optimal plugin design that just cannot hide Pd as the regular way of compiling plugins with externals, we still need to test and measure if this process has also some technical disadvantages, such as quantifying the increase in latency that this way of loading externals into camomile causes⁷. But one way or another, the possibility to have any externals available like that is very promising and exciting.

One disadvantage with this workaround is the increase of the generated plugins' memory size, once each plugin needs to have a copy of Pure Data and the externals needed embedded in the plugin. To mitigate this issue, it is possible to install Pd and the externals manually on the machine in order to use the plugins. But then, this includes yet more steps and it is definitely not something you want for a regular plugin user that never heard of Pd. Though it is an option for the Pd enthusiasts.

But since the amount memory we're talking about is not that significant these days, the first workaround can be the usual way to easily include externals into camomile without doing what we're providing here with camomile-else.

5. Conclusion

Pure Data is a cross platform visual programming environment for interactive computer music and multimedia with a friendly visual paradigm that is well suited for artists and non programmers. For this reason, it made the world of computer music more accessible and increased the range of users capable of building sounds and compose music with computers.

Pd also provides support to create smartphone apps, create strange devices and installations with microcomputers like Raspberry Pi, new instruments with devices like Bela and so on. Its portability and accessibility is unparalleled. The integration of Pd and any plugin host is now also a reality, which brings the Pd world into the workflow of DAWs and sound editors.

This is true for either the Pd enthusiast that can now easily also work with Pd inside plugin hosts but it also a way for non programmers to design and distribute plugins with Pd patches (without a background in programming or computer science). This opens the doors for the commercialization of open source plugins and a fertile terrain to explore and research computer music and experimental techniques in for yet a broader range of users.

This is where our work with Camomile-ELSE has a big contribution as it allows users to design plugins with an extensive set of externals provided by the ELSE library, as well as the extensive examples available in its included tutorial.

This development effort also opened us to new aims like provide better accessibility in the creation of mobile apps by also compiling ELSE into other libpd projects such as PdDroidParty and PdParty.

Another development from this work would be a basis for other projects to also include other external libraries.

⁶ See: <https://web.archive.org/web/20210711175310/https://github.com/pierregrillot/Camomile/discussions/256>

⁷ See: <https://web.archive.org/web/20210711175710/https://github.com/pierregrillot/Camomile/discussions/233>

References

- [1] PdParty see:
<https://web.archive.org/web/20210711154116/http://danomatika.com/code/pdparty>
- [2] PdDroidParty see:
<https://web.archive.org/web/20210410181502/http://droidparty.net/>
- [3] LibPd see:
<https://web.archive.org/web/20201125151522/http://github.com/libpd/libpd>
- [4] ELSE Library see:
<https://web.archive.org/web/20210711155857/http://github.com/porres/pd-else>
- [5] Camomile-ELSE see:
<https://web.archive.org/web/20210711180229/http://github.com/emviveros/Camomile-ELSE>
- [6] Porres, Alexandre Torres. “ELSE Library for Pure Data”, 2017, 8. See:
<https://web.archive.org/web/20210711160308/http://compmus.ime.usp.br/sbcm/2017/papers/sbcm-2017-6.pdf>
- [7] PD Pulp see:
<https://web.archive.org/web/20201207051840/http://github.com/logsol/Pd-Pulp>
- [8] JUCE see:
<https://web.archive.org/web/20210708055657/http://juce.com/>
- [9] Camomile see:
<https://web.archive.org/web/20210711163045/http://github.com/pierreguillot/Camomile>

Web Art em tempos de HTML5

Avner Maximiliano de Paulo¹, Luiz Gustavo Colzani Monti Sousa¹,
José Mauro da Silva¹, Flávio Luiz Schiavoni¹

¹ ALICE - Arts Lab in Interfaces, Computers, and Everything Else
Departamento de Ciência da Computação - Universidade Federal de São João Del Rei

avnerpaulo.mg@gmail.com, lgustavocms@gmail.com, jmsandy@gmail.com, fls@ufsj.edu.br

Resumo. Este artigo tem como objetivo explorar como se deu o início da arte digital, como surgiu a web art, como se produz a web art hoje e quais tecnologias desenvolvidas a partir do ideal artístico oferecem possibilidades para a arte e outras áreas.

Abstract. This article aims to explore how was the beginning of digital art, how web art came about, how web art is produced today and what technologies developed from the artistic ideal offer possibilities for art and other areas.

1. Introdução

A criação artística apoiada por computadores é talvez quase tão antiga quanto a própria computação. Desde os primórdios da computação artistas vem se utilizando de computadores como ferramentas de apoio para suas criações artísticas tradicionais. Isto inclui os diversos campos das artes como a fotografia, pintura, teatro, dança, música, arquitetura, escultura e literatura, por exemplo, campos que foram radicalmente impactados pelo surgimento de apoio computacional. Com o tempo, o computador passou de ferramenta de apoio às artes tradicionais (das chamadas Belas Artes até as chamadas “Artes Menores”, como o artesanato) para uma ferramenta de criação específica e não amarrada às artes convencionais. Isto permitiu o surgimento de um outro tipo de arte, possível apenas por meio da computação, chamada Arte Digital[1].

A Arte Digital surge como uma convergência entre Arte, Ciência e Tecnologia e trouxe para os artistas novos desafios para a criação artística no universo digital [2]. Esta Arte Digital inclui diversas novas formas como a Media Art, a Arte em rede (net art), as instalações digitais e a Realidade Virtual como forma de Arte [1]. Com o surgimento da Internet, a web passou a despertar o interesse de artistas, como forma de distribuir a arte tradicional, por meio de lojas, museus e galerias de arte virtuais, e também como forma de fazer arte por permitir a criação multimídia e estar presente no centro na revolução tecnológica da informação [3] [4]. Esta forma de arte, chamada **Web Arte**, inclui diversas criações artísticas feitas especialmente para a web [4].

Um aspecto importante na web arte está relacionada a seus processos e ferramentas. Algumas outras formas de Arte digital dependem da aquisição de equipamentos, tecnologias e ferramentas de alto custo de aquisição para a criação artística. Já a web arte possui um baixo custo para sua produção já que o HTML é uma linguagem gratuita, o HTTP é um protocolo gratuito e os navegadores são ferramentas gratuitas na maioria das plataformas.

Em geral, não são necessárias ferramentas especiais para a criação nesta plataforma, sendo necessário apenas um editor de texto, o que torna esta arte também muito simples de ser compartilhada. As demais tecnologias associadas a este fazer artístico, como servidores web, bancos de dados e linguagens de programação *client-side* ou *server-side* também possuem versões gratuitas. Isto faz com que este formato de arte seja amplamente acessível para artistas digitais [3].

Desde os primórdios da Arte Digital, a criação artística passa pela criação de aplicações desenvolvidas especificamente para o fazer artístico onde as obras artísticas são programas de computador. No caso da web arte, estes programas são páginas web que diferem de páginas tradicionais por possuírem propósitos artísticos, muitas vezes explorando a capacidade das páginas para além do que elas foram projetadas. Logo, a evolução da criação artística nesta plataforma passa pela evolução de capacidade de representação das páginas web, a criação de novas ferramentas, linguagens de programações, API's e frameworks para o desenvolvimento de aplicações deste domínio específico. No centro da evolução da web está atualmente o HTML5, que trouxe para a criação de arte na web diversas novas possibilidades, que inclui a o processamento de som e imagens, a síntese de imagem, som e vídeo e a comunicação em rede.

2. Da Arte Digital para a Web Arte

O fortalecimento dos ideais da arte pós-guerra ou arte contemporânea, que rompe com as regras e tradições do modernismo, em conjunto com o advento da computação na segunda metade do século XX, dá início a projetos artísticos que visavam promover a colaboração da arte com as tecnologias recém-criadas [5]. Também nesta época é dado o início da Arte Digital como uma forma de arte produzida com a utilização de software e hardware e distribuída em meios digitais, tendo como produto artístico obras multimídias que levem ao espectador a experienciar criações visuais e sonoras, não necessariamente separadas [5].

O avanço da computação e o surgimento de vários meios de comunicação com apoio tecnológico proporcionou à Arte Digital um avanço em vários âmbitos, sendo um dos mais impactantes as técnicas de interação com o espectador, que o convida a participar da obra [6]. Possivelmente como uma herança dos jogos de computadores e das pesquisas com interfaces, esta interatividade permite uma quebra de papéis entre artista e público de modo que o espectador deixe de ser somente a parte que assiste a obra

criada e participe da proposta do artista [7]. Esta interatividade, tão explorada em instalações artísticas tecnológicas, também habita o universo da web arte e permite ao artista levar uma experiência interativa para os visitantes de um site.

No início da web arte os artistas tinham pouco recursos para criar suas obras. Links, texto, imagens e animações eram o que existia e tal limitação provocou a criatividade de muitas pessoas que utilizaram esta limitação como fator estético para criar sites artísticos. Com isto, muitas obras seguiam estéticas puramente digitais como a tendência Glitch de se aproveitar de erros de computadores para a criação artística [8, 9]. Isto gerou uma série de obras de web arte, páginas web, que não tinham outra função se não despertar a curiosidade do internauta por razões estéticas e não funcionais.

3. A tecnologia do HTML 5

Se a web e as primeiras versões do HTML, com sua escassez de recursos, permitiram o surgimento da web arte, a evolução da mesma trouxe consigo um conjunto de ferramentas que permite criações artísticas diversas. Este conjunto de ferramentas está disponível no HTML5, o que faz dele o meio mais avançado de produção de web arte hoje.

Música: A web sempre foi uma construção multimídia, permitindo inserir e ligar arquivos com música em páginas HTML. No entanto, com o advento do HTML5, surge a API **webaudio** que transforma o navegador em um sintetizador para a criação e processamento de áudio em tempo real[10]. A API WebAudio permite trabalhar com diversas fontes sonoras, de arquivos à entrada de microfone do computador e gerar resultados sonoros com processamentos no próprio navegador além de poder ser utilizada, junto com as API's de imagem, para criação de jogos, animações com sons específicos e composições, como o web violão apresentado na Figura 1¹. Juntamente com o WebAudio, surgiu a API **WebMidi**, que trouxe ao HTML5 a possibilidade de conectar periféricos musicais que utilizam o protocolo MIDI de comunicação e permite, por meio de um mapeamento definido pelo artista, controlar alguma aplicação desenvolvida por ele.

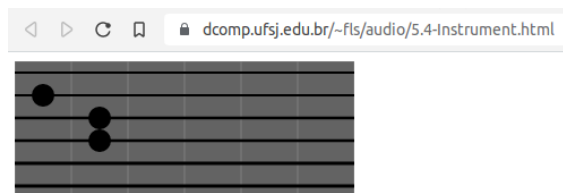


Figura 1: Web Violão criado com SVG e webaudio.

Artes Visuais: Na área de artes visuais, o HTML5 trouxe uma API chamada **Canvas**, um componente gráfico que foi inicialmente introduzida pela Apple para usar no Mac OS X WebKit component em 2004. Atualmente,

¹Disponível em <https://dcomp.ufsj.edu.br/~fls/audio/5.4-Instrument.html>.

esta API tem a capacidade de desenhar gráficos, imagens, criar animações, processar e renderizar vídeos em tempo real através de código. O canvas permitiu o surgimento do **WebGL**, tecnologia baseada no OpenGL ES 2.0 que permite ao artista, sem a utilização de nenhum plug-in, fazer renderização de objetos 3D em um espaço criado pelo Canvas no HTML utilizando para isto a GPU da máquina. Além do Canvas, o HTML5 trouxe uma possibilidade de interagir com desenhos vetoriais por meio da API **SVG** ou Scalable Vector Graphics, como apresentado na Figura 2². Estas imagens vetoriais em 2D ou 3D utiliza uma linguagem de marcação que representa pontos onde o navegador deve desenhar linhas e também uma API em Javascript para manipular estas imagens em tempo real.

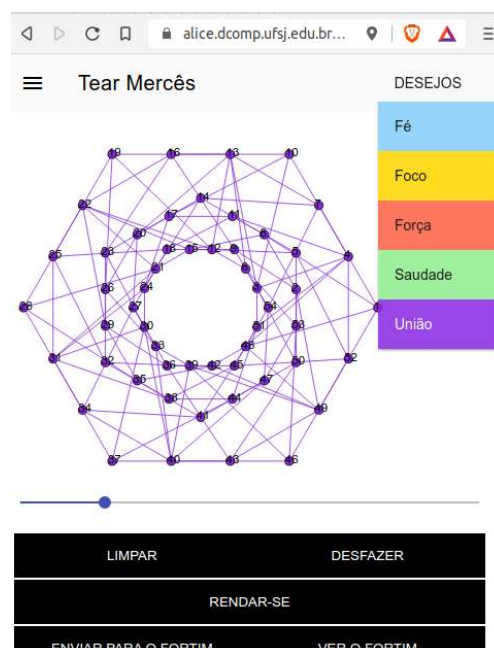


Figura 2: Tela de criação baseada em SVG do projeto Renda-se.

Interatividade: O HTML5 trouxe também uma API que permite que seja feita a conexão entre o navegador e o servidor, chamada **WebSocket**, sem que o cliente necessite fazer uma chamada de nova página para isto, possibilitando ao artista desenvolver páginas capazes de interagir sozinhas às escolhas do espectador e que mudam de forma de acordo com essas escolhas, com ou sem a intervenção de outras pessoas. Além da conexão de rede, o HTML5 também trouxe a possibilidade de acesso e conexão aos dispositivos de mídia como microfone e câmeras por meio da API **WebRTC**. Esta API possui integração com o CSS e com o WebGL e permite a criação de filtros para imagens e vídeos em tempo real. Em dispositivos onde estes sensores estão presentes, o HTML5 permite o acesso à localização do aparelho, por meio da Geolocation API, o acesso ao acelerômetro, giroscópio e bússola para a captura de gestos, o que pode aumentar ainda mais a capacidade de interação com estas páginas.

²Disponível em <https://alice.dcomp.ufsj.edu.br/webart/tear.html>.

4. Web Arte em tempos de HTML5

As API's citadas anteriormente permitem a criação de obras de web arte mais complexas do que as feitas nos primórdios da web arte, como as mostradas em nossas Figuras. Contudo, tais API's não são necessariamente voltadas para a criação artística e o surgimento de tais API's permitiu o desenvolvimento de diversos *frameworks* e ferramentas que integram estas tecnologias simplificam a criação de web arte. Com isto, algumas formas de Arte Digital como a Criação de instrumentos musicais digitais, a Realidade Ampliada e o *Live Coding* puderam ser incorporadas ao universo da web arte. Estas formas de não arte não são as únicas que se tornaram possíveis com o HTML5 mas ilustram a evolução da web como plataforma para a criação artística a partir desta tecnologia.

Criação de Instrumentos musicais: A criação de novos instrumentos musicais é uma área de pesquisa para a Arte Digital e a Música contemporânea. Algumas ferramentas e *frameworks* foram criados para um público de artistas e músicos com a proposta de resolver problemas comuns de arquitetura, enfrentados quando lidando com ambientes de música de computador, tendo como objetivo possibilitar um melhor entendimento da lógica dos instrumentos digitais. Exemplos são os *frameworks* Flocking.js[11], Tone.js [12], Tune.js [13] e Timbre.js [14].

Realidade Ampliada: A Realidade Ampliada é a adição de elementos de realidade virtual em um ambiente real. A capacidade de utilizar a webcam do computador ou celular e processar os dados desta permitiram que a realidade ampliada fosse possível no HTML. Exemplos de *frameworks* que permitem esta integração são o Awe.js e o AR.js[15].

Live Coding: A codificação em tempo real é a escrita de código para a criação artística feita ao vivo em eventos voltados para este fim. Neste tipo de criação artística digital, escrever código é parte da performance artística e os artistas projetam suas telas para que o público possa acompanhar suas codificações[16]. O HTML5 permitiu o surgimento de ambientes de *live coding* para navegadores que conta com a utilização de várias API's como Canvas, WebAudio e WebGL. Exemplos destas tecnologias são Gibber [17], Wavepot, McLuhan.js[18] e Lissajous.js.

5. Considerações sobre o ciberespaço como lugar de produção artística

Atualmente, a Internet ocupa um grande espaço da nossa sociedade, espaço este que envolve transações comerciais, relações humanas, troca de informações, divulgação de conhecimento, entre outras. A internet nos fez mudar nossa posição de agir e pensar como seres humanos, ocupando grandes áreas de conhecimento criadas por nós. Este novo espaço criado pela ocupação da internet, é chamado de ciberespaço, uma "gênese de um novo mundo". O estudo do ciberespaço encontra-se em teorias pós-modernas, visto que no campo de estudo moderno, conceitos como imaterialidade e desterritorialização não são contemplados. A

partir do ciberespaço, surge então, a cibercultura que define conjunto de formas de expressão, de atitudes, de modos de pensamento e de valores que se desenvolvem lado a lado ao crescimento do ciberespaço.

Com o consumo da internet à várias áreas humanas, seria difícil não pensar na arte sem a absorção dessas novas tecnologia. Diante disso, surge a Web Arte, um estilo de arte que desde dos primórdios até a atualidade é marginalizado tanto no mercado artístico como na definição de arte em si. Nos primórdios da Internet, não havia uma adequação do ambiente para a produção artística em si, mas essa barreira jamais impediu que produções artísticas surgissem no ciberespaço.

Observando o passado percebemos que a tecnologia sempre proporcionou aos artistas novas ferramentas para que expressassem sua arte. Nos tempos atuais, mantém-se a mesma ligação entre estas duas áreas, contudo, a tecnologia hoje em dia é também um motor para o desenvolvimento de novas formas de arte [19]. Desta maneira, avançar tecnologicamente significa avançar nas formas de se produzir arte. Assim, com o desenvolvimento de ferramentas e modernização da Internet juntamente com o pensamento artístico dentro da rede, houve uma capacitação e adequação do ambiente para que artistas conseguissem de forma mais simples e clara realizar produções artísticas complexas dentro do ciberespaço.

A produção de arte não tem que ser algo custoso, como é ambientado em diversas produções artísticas tradicionais. A presença de novas tecnologias populares e baratas nos fazem repensar a forma como vemos a produção artística digital. Desde a década de 1960, com o movimento da Arte *Povera* que repensava o modo de se fazer arte, o conceito tradicional e vanguardista da produção artística tem sido debatido. Este movimento procurava derrubar barreiras da produção artística para as pessoas mais pobres, fazendo-as repensar a arte e alimentar uma crítica ao capitalismo e a produção desenfreada de bens. Este movimento também alcançou a web, o que nos faz pensar em uma arte digital *Povera* que permite uma produção artística que é digital e barata, usando dos princípios do próprio movimento da década de 60, como o uso de computadores antigos, com baixo poder de processamento, equipamentos domésticos, além de ferramentas de baixo custo ou sem custo algum, como é o caso das tecnologias aqui apresentadas.

Essa forma de criar e promover arte, de forma simples e barata, pode abranger grupos muito grande de pessoas, sem a necessidade do deslocamento para apreciar e participar das obras, trazendo a questão social da arte para o debate. Ao invés de deslocar pessoas para espaços de fruição de arte, deslocamos a arte até o espaço pessoal, seja ele a tela de computadores pessoais ou de aparelhos celulares. Esta inversão de movimentação que permite que a arte invada os dispositivos pessoais e permite que ela rompa fronteiras ou horários de funcionamento para sua atenção, permitindo a interação de pessoas que dificilmente iriam interagir sem este aparato tecnológico.

6. Conclusão

Este artigo tentou apresentar o caminho da produção de web arte apresentando o tipo de arte cuja principal meio / ferramenta para produção, desde a origem até a atualidade, é o HTML e as páginas web. Esta forma de arte, ainda muitas vezes questionadas devido a sua transdisciplinaridade, é uma vela que impulsiona a criatividade, a ciência e o desenvolvimento tecnológico, partindo de uma apropriação estética que possui apenas a finalidade artística, e chegando a diversas aplicações que possuem um viés acadêmico / mercadológico / científico que não inclui necessariamente uma discussão estética.

Apesar de existirem ferramentas, frameworks, API's e linguagens de programações que abarcam as áreas da música, artes visuais e interatividade, há muito o que ser produzido ainda, especialmente se tomarmos como ponto de partida critérios de sustentabilidade em arte onde busca-se que a mesma seja amplamente difundida e não apenas restrita a uma parte da população [20]. Esta é uma das características marcantes da web arte, uma área relativamente nova que possui muito a ser debatido. O fato de estar amplamente difundida por tecnologias livres faz da Web Arte, e consequentemente do HTML5, ferramentas artísticas viáveis e acessíveis, aspectos tão importantes para a difusão da arte com tecnologia.

No princípio, a web arte, com toda a limitação tecnológica que esta mídia possuía, já a tornava algo a ser utilizado. A evolução das tecnologias a tornaram ainda mais especial. Sempre existiram ferramentas capazes de tratar imagem e som, como Canvas e o SVG, que eram relativamente fáceis de utilizar, mas cuja difusão dependia da instalação de aplicações especiais para este fim. Esse fato dificultava bastante a disseminação de conteúdo artístico, já que nem todos dispunham dessas ferramentas. Com a chegada do HTML5 iniciou-se uma nova etapa na criação artístico digital uma vez que ele foi capaz de unir todas essas API's e permitir criar novas ferramentas que possibilitem novas formas de fazer web arte. Podemos assim constatar a importância que o HTML atingiu, surgindo como uma simples linguagem de Markup e evoluindo para uma linguagem que atende propósitos artísticos capaz de criar aplicações complexas de uso geral.

Tentar prever um possível futuro para as tecnologias nascidas desse meio é um exercício que requer um esforço enorme de imaginação e que certamente poderá falhar. No entanto, acreditamos que este esforço é o que permitirá a realização deste futuro e o surgimento de novas tecnologias. O que podemos ter certeza, nos dias de hoje, é que esta forma de arte está mais presente do que nunca instigando as pessoas a ver o mundo de outra forma e usando para isto, seus navegadores de Internet.

7. Agradecimentos

Os autores gostariam de agradecer os membros do Laboratório ALICE (Arts Lab in Interfaces, Computers, and Everything Else) da Universidade Federal de São João Del

Rei, um grupo interdisciplinar que envolve alunos de cursos como Ciência da Computação, Música, Artes Cênicas, Visuais e Aplicadas e que é a base fundamental do debate que gerou esta pesquisa. Os autores gostariam de agradecer o apoio financeiro institucional do CNPq, FAPEMIG e da PROPE/PROAE/UFSJs.

Referências

- [1] Bruce Wands. *Art of the Digital Age*. Thames & Hudson, 2007.
- [2] Oliver Grau. *Virtual Art: from illusion to immersion*. MIT press, 2003.
- [3] Mark Tribe, Reena Jana, and Uta Grosenick. *New media art*. Taschen London and Cologne, 2006.
- [4] Annette Weintraub. Art on the web, the web as art. *Commun. ACM*, 40(10):97–102, October 1997.
- [5] saga.art.br. Você sabe o que é arte digital? <https://saga.art.br/voce-sabe-o-que-e-arte-digital/>, 2016. Accessed: 2018-05-21.
- [6] Jennifer Seevinck. Interaction in art and computing. In *Springer Series on Cultural Computing*, pages 31–46. Springer International Publishing, 2017.
- [7] Linda Candy. Evaluation and experience in art. In *Interactive Experience in the Digital Age*, pages 25–48. Springer, 2014.
- [8] Jonas Downey. Glitch art. *Ninth Letter*, 118, 2002.
- [9] Rosa Menkman. Glitch studies manifesto. *Video vortex reader II: Moving images beyond YouTube*, pages 336–347, 2011.
- [10] Paul Adenot, Chris Wilson, and Chris Rogers. Web audio api. *W3C, October*, 10, 2013.
- [11] Colin BD Clark and Adam Tindale. Flocking: a framework for declarative music-making on the web. In *SMC Conference and Summer School*, pages 1550–1557, 2014.
- [12] Yotam Mann. Interactive music with tone.js. In *Proceedings of the 1st annual Web Audio Conference*. Citeseer, 2015.
- [13] Benjamin Taylor and Andrew Bernstein. Tune.js: A microtonal web audio library. 2016.
- [14] Charles Roberts, Graham Wakefield, and Matthew Wright. The web browser as synthesizer and interface. In *NIME*, pages 313–318. Citeseer, 2013.
- [15] Daeil Seo, Byounghyun Yoo, and Heedong Ko. Webizing mixed reality for cooperative augmentation of life experience. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work and Social Computing Companion*, pages 401–404. ACM, 2016.
- [16] Nick Collins, Alex McLean, Julian Rohrerhuber, and Adrian Ward. Live coding in laptop performance. *Organised sound*, 8(3):321–330, 2003.
- [17] Charles Roberts. Code as information and code as spectacle. *International Journal of Performance Arts and Digital Media*, 12(2):201–206, 2016.
- [18] Benjamin Francis Taylor. Mcluhan.js: Live net art performance with remote web browsers. 2016.
- [19] Eyal Gever. Technology and art: Engineering the future. <http://www.bbc.com/news/entertainment-arts-19576763>, 2012. Accessed: 2018-05-26.
- [20] Adilson Roberto Siqueira. Arte e sustentabilidade: argumentos para a pesquisa eco-poética da cena. *MORINGA-Artes do Espetáculo*, 1(1), 2010.

Cerâmica e silício na criação musical

Carlos Eduardo Oliveira de Souza^{1, 4}, Jônatas Araújo da Silva^{2, 4},
Rafael Alves Soares de Andrade^{1, 4}, Flávio Luiz Schiavoni^{3, 4}

¹ Departamento de Música

² Departamento de Artes Aplicadas

³ Departamento de Computação

⁴ Arts Lab in Interfaces, Computers, and Everything Else - ALICE

Universidade Federal de São João del-Rei (UFSJ)

São João del-Rei, MG – Brasil

carlosbolin@aluno.ufsj.edu.br, rafael.asa@outlook.com

jonatas25araujo@gmail.com, fls@ufsj.edu.br

Resumo. *Esse artigo aborda uma pesquisa transdisciplinar entre áreas de música, computação, acústica e cerâmica e apresenta a criação de um instrumento musical digital composto por diversos smartphones que se comunicam por protocolos de rede gerados em patches criados no Pure Data em conjunto com a plataforma MobMuPlat. Neste instrumento temos um conjunto de "vozes", onde cada celular exerce o papel de um músico abstrato. Esses celulares são acoplados em amplificadores modelados em cerâmica que respondem de formas diversas às frequências tocadas, criando assim, uma camada sonora alterada, compondo o instrumento não só como um amplificador, mas também como efeito, influenciando no timbre final.*

Abstract. *This paper addresses a transdisciplinary research between areas of music, computing, acoustics and ceramics and presents the development of an instrument composed of several smartphones that communicate through network protocols generated in patches created in pure data in conjunction with the MobMuPlat platform. Therefore, we have a set of "voices", where each cell phone plays the role of an abstract musician. These cell phones are coupled in amplifiers modeled in ceramic that respond in different ways to the frequencies played, thus creating an altered sound layer, composing the instrument not only as an amplifier, but also as an effect, influencing the final timbre.*

1. Introdução

Com a evolução da tecnologia móvel, os aparelhos celulares se tornaram computadores portáteis presentes na vida cotidiana de muitas pessoas. O objetivo inicial destes dispositivos era a comunicação e hoje os mesmos cumprem este papel ultrapassando a simples comunicação de voz e permitindo a comunicação por outras mídias. A expansão de tais limites comunicacionais foi possível graças à evolução da capacidade dos dispositivos, principalmente no que tange à conectividade e à capacidade de processamento aliada à possibilidade de manipulação de materiais multimídia.

A partir desta expansão surgiu a possibilidade de implementação de aplicativos que transformam dispositivos móveis em instrumentos musicais e vários aplicativos

musicais encontram-se disponíveis para este fim. Junto com isto, surgiu a possibilidade de realizar música em conjunto utilizando dispositivos móveis como instrumentos musicais digitais (IMD). A prática musical em conjunto por meio de dispositivos computacionais já é popular, como é o caso das Orquestras de computadores portáteis ("Laptop orchestra" - LOrc) e orquestras de celulares. A utilização de aparelhos celulares para práticas musicais é vista hoje como a evolução das orquestras de computadores portáteis devido à simplicidade, capacidade e praticidade de tais dispositivos.

1.1. Amplificando uma orquestra de celulares

Apesar de os dispositivos móveis trazerem muitas possibilidades para a criação musical, os mesmos ainda possuem uma grande limitação que é sua baixa potência sonora. A solução encontrada para esta questão por diversos grupos de música e LOrcs foi criar amplificadores eletrônicos para aumentar a potência sonora dos dispositivos móveis. Estes projetos permitem que o som do IMD possa ressonar no espaço de concerto partindo do lugar onde o músico se encontra, permitindo ao público identificar as fontes sonoras de uma forma similar ao que aconteceria com instrumentos acústicos.

Apesar de ser bastante eficaz, a amplificação eletrônica pode trazer outras questões para um projeto de criação musical utilizando celulares como a) a necessidade de conhecimento em eletrônica para a criação de tais dispositivos, b) a necessidade de um laboratório e equipamentos para a confecção do mesmo, e c) o alto custo de materiais que este projeto pode trazer. Além disto, d) estes equipamentos trazem a necessidade de alimentação elétrica para os mesmos, o que é um fator limitante para a utilização dos mesmos em espaços abertos como parques ou ruas da cidade. Tentando contornar estes problemas e solucionar o problema da baixa potência sonora dos dispositivos móveis, procuramos encontrar a solução deste problema por meio da amplificação acústica destes dispositivos.

1.2. A solução proposta: amplificadores cerâmicos

Os aparelhos celulares possuem alto-falante embutidos sendo, no entanto, desprovidos de uma concha para amplificar acusticamente o seu som. Em nosso trabalho, nos

baseamos na possibilidade de criação de ressonadores que permitem a amplificação de som por meio de acoplamento de ondas sonoras, utilizando este conceito para a criação de amplificadores acústicos para aparelhos celulares. Este material foi escolhido para nosso trabalho por ser fácil de modelar e também por permitir grandes variedades de formas, texturas, queimas, e acabamentos.

A ligação entre a acústica e a cerâmica é antiga e nos remete algumas práticas, como os ressonadores do teatro grego. Na época, para auxiliar na amplificação, eram usados vasos ressonadores em cerâmica embutidos nos degraus do auditório, os chamados ressonadores de Helmholtz. Nestes ambientes, o som gerado no palco passava por um sistema acústico de cerâmica agindo como uma mola que é comprimida e rarefeita gerando a amplificação de faixas que coincide com a frequência da voz humana. Estes e outros conceitos foram utilizados na criação de nossos amplificadores, apresentados na Seção 2.2.

1.3. As questões tecnológicas

Além da questão da amplificação acústica, nosso projeto também se debruçou sobre a questão tecnológica da criação musical para aparelhos celulares em grupo. Assim como em trabalhos que utilizam a criação de uma interface, que após ser acoplada a um objeto físico compõe um instrumento digital [1], optamos por utilizar uma solução baseada em tecnologias conhecidas e fundamentadas, como o Pure Data e o aplicativo MobMuPlat, utilizando as redes de computador para sincronizar os dispositivos e a acústica do ambiente para mesclar os sons. Mais sobre a questão tecnológica será apresentada na Seção 2.1.

2. Cerâmica, Silício e criação musical

O primeiro passo deste projeto foi definir as atribuições e iniciar as pesquisas tomando dois caminhos: a construção de uma interface de software que permitisse a sincronização em rede dos dispositivos utilizados na apresentação e a construção de amplificadores cerâmicos que auxiliassem na amplificação do som destes dispositivos.



Figure 1: Maestro e vozes acopladas aos amplificadores de cerâmica

2.1. A Interface de software

A interface de software de nosso ambiente musical foi montada utilizando o ambiente de programação musical

Pure Data como engine de som e o aplicativo mobile MobMuPlat para a interface gráfica e também para integrar os patches do Pd nos celulares. Os objetivos iniciais deste meta-instrumento [2] foram trazer diversos conceitos de protocolos de redes de forma a favorecer a programação do instrumento, onde dispositivos diferentes exercem papéis diferentes na comunicação, compondo uma subpesquisa sobre as melhores formas para sincronizar esses instrumentos, de forma que trabalhem em tempo real [3]. Para esta distribuição de papéis, utilizamos a metáfora da orquestra / coral de celulares, com papéis como maestro e músicos / vozes, onde os dispositivos em cena podem possuir uma determinada voz e um dispositivo fora de cena irá assumir o papel de maestro, atuando na distribuição de tarefas e sincronização das vozes.

Optamos por criar um patch único para as “vozes” de nosso coral de dispositivos móveis, onde é possível selecionar em qual porta o celular vai se conectar. Assim, todos executam o mesmo patch, mas recebem mensagens diferentes de acordo com a porta selecionada permitindo que os diferentes dispositivos tenham diferentes comportamentos musicais. Esse seletor aparece na tela inicial da interface feita no MobMuPlat.

A ferramenta das Vozes, criada no Pd, possui vários dispositivos de gravação e efeitos para permitir composições que utilizam os sons do ambiente e as modificações destes sons na sua reprodução. Além de gravadores e efeitos, estão presentes efeitos de ruído branco e comandos para realizar ações com as funcionalidades físicas do celular, como vibracall e lanterna, além do som que imita o toque de um telefone fora do gancho.

Na parte dos efeitos, foram implementados seletores de liga e desliga para controlar a atuação dos mesmos. Foi implementado uma modulação em *anel* e *delay*, sendo estes presentes no mesmo esquema, alterando somente valores de entrada para a escolha entre um ou outro. Esses efeitos atuam na execução do trecho de áudio armazenado em um *buffer* de gravação.

Além desse, há outro patch para a representação do “maestro”, que envia comandos para os celulares tocarem e executarem suas ações. Essa comunicação é realizada por comandos nativos do Pd, tais como o *send* e o *receive*, que em conjunto formam uma rede de comunicação entre maestro e vozes. Para a mensagem ser enviada para outro dispositivo, fazemos uso do patch chamado *PdWrapper*, que cria todos os protocolos e conversões necessárias para realizar a ponte entre um dispositivo e outro baseando-se no formato *OSC* para o encapsulamento das mensagens. Em princípio, o maestro está sendo executado em um *notebook* e há uma partitura em *OSC* feita para agrupar os comandos definidos para as vozes e criar peças utilizando esta estrutura. Na Figura 1 mostramos o maestro e dois celulares executando suas vozes.

2.2. Amplificadores Cerâmicos

O processo de criação dos amplificadores ocorreu em ateliê de cerâmica, usando argilas com propriedades plásticas e de alta temperatura. Esta argila, além de oferecer a

possibilidade de aguentar altas temperatura, possibilita a utilização de vidrados cerâmicos com a finalidade de mudar a textura das paredes internas da peça. As peças foram modeladas em um torno de cerâmica para conseguir uma variedade as formas, praticidade de acabamento e uniformidade nas paredes das peças. Uma vez a peça modelada, esperamos o ponto de couro (peças parcialmente secas) para o acabamento manual. Nessa etapa foi feito um corte onde se encaixa o celular. Muitas vezes as peças são polidas para os primeiros testes, e em algumas peças foi preciso fazer suportes.

Após a secagem das peças fazemos os primeiros testes de amplificação nos laboratórios da universidade. Usamos ruído branco para a comparação de amplificação e também gravações de Freq Sweep com e sem os amplificadores para verificar quais frequências eram acopladas, quais seriam canceladas e medir o resultado final da amplificação acústica em uma escala de decibéis.

Após a primeira etapa de criação e gravação, foi feita a primeira queima, que chamamos de biscoito. Essa queima serve para tirar toda a água presente na peça. Nesse processo, a peça fica mais resistente e encolhe um pouco. Após o processo da queima de biscoito foi feito mais um teste de amplificação do som. Nesse teste a amplificação fica mais aparente já que a peça está totalmente sem água e mais porosa.

Por fim, temos a queima de alta temperatura, etapa na qual temos a possibilidade de esmaltar a peça para a vitrificação. Além disso, podemos escolher um esmalte cristalino, fosco e com textura. São muitas variedades, porém ainda não chegamos na etapa de queima de alta temperatura das peças.

Além da variedade de esmalte, após a última queima as peças são sinterizadas e geralmente encolhem em até 25 % do seu tamanho quando modelado. Há ainda uma variedade de queimas de alta temperatura, tais como, queima no forno noborigama, forno lenha, forno elétrico, e forno a gás. São muitas opções com variantes que podem influenciar na amplificação final das peças. Acreditamos que alguns amplificadores podem atuar apenas em uma faixa de frequência sendo um desafio conseguir fazer um amplificador que consiga amplificar o máximo de faixas de frequência possível.

Com os testes todos prontos, conseguimos analisar e comparar a capacidade de cada amplificador, levando em consideração seu formato, tamanho e diâmetro da boca. Podemos ainda estudar qual chega perto de amplificar mais faixas de frequência.

3. Resultados Iniciais: O telefone sem fio

Um primeiro resultado vindo desse processo de criação, além da própria pesquisa colaborativa (que será mencionada logo abaixo), foi a criação da peça “Telefone sem fio”, nome criado a partir de uma analogia com a brincadeira infantil de passar uma mensagem de uma criança para outra até que a mensagem seja totalmente perdida.



Figure 2: Primeiros protótipos dos ressonadores.

Inspirada em outras peças que utilizam a mesma tecnologia [4], esta peça traz celulares conectados via rede e cria uma reflexão a partir da perda de informações, trazendo para a discussão o problema das chamadas fake news. A peça consiste em gravar samples de áudios, declamados pelo performer e essas gravações vão sendo reproduzidas pelos dispositivos, com a adição de efeitos, distorções e manipulações no áudio, onde no final temos uma figura sonora muito diferente da original, dando sentido ao conceito.

Nesta peça, são acionados diversos aparelhos em ordens alternadas ou simultâneas, criando uma camada visual com as telas acendendo e desligando, para além da camada sonora presente a partir dos toques acionados e da reprodução dos samples manipulados. A peça foi dividida em seis momentos, começando com os celulares vibrando e acendendo seus flashes, sendo esta uma introdução para o seguinte movimento onde são declamadas as frases que servem de material sonoro para a composição dos samples que serão modificados.

Partindo para o próximo movimento da peça, trazemos a reprodução desses samples, que vão sendo atrasados em cada falante, criando assim um deslocamento de fase, que evolui para a terceira parte. Nesta parte, um celular toca enquanto outros gravam, resultando em uma perda de informação quando executado. Quem está gravando tem tela em vermelho piscante e quem está falando toca com uma tela preta. Neste momento, não temos mais o som do vibracall e os dispositivos falam ao mesmo tempo, com vozes entrecortadas como se quisessem vencer uma batalha sonora pelo controle de espaço acústico, partindo para o quarto movimento. Esse é o momento de maior saturação da peça, com todos os efeitos ligados.

Quando este movimento se encerra, com uma transição para a quinta parte, as vozes vão se silenciando uma a uma, restando apenas as vibrações e os flashes, que quando param, direcionam para o fim da peça. Por fim, o sexto movimento, traz apenas um sinal de um telefone fora do gancho, tocando um lá afinado em 440hz, característico dessa figura.

4. Discussão

Devido às limitações enfrentadas durante o período de pandemia, não foram realizados todos os testes propostos em nossos amplificadores. Também não foi possível testar outras composições para nossa estrutura já que a mesma de-

pende fortemente da utilização dos amplificadores e nossa equipe está atualmente dispersa geograficamente em diferentes cidades. Também não foi possível estreitar esta peça e ter um feedback do público. Com isso ainda faltam alguns dados que seriam coletados a partir dessas experiências, tornando o desenvolvimento da pesquisa mais lento que o esperado. Tais testes envolviam a análise do funcionamento do instrumento com mais dispositivos conectados, inclusive durante uma performance, acoplando os celulares dispostos pelo público participante.

A análise dos samples de áudio gravados a partir de cada falante, para a verificação da alteração nas frequências proferidas por cada um também foi interrompida, devido a falta de acesso aos laboratórios da universidade, sendo este, um outro ponto para ser continuado futuramente.

A intenção desta peça é permitir a participação do público a fim de criar uma dinâmica trabalhando o conceito de happening. No presente momento, com o impedimento da circulação de possíveis trabalhos artísticos apresentados com uma devida instalação de forma presencial, o trabalho se encontra em uma forma estática de apresentação, apenas com o material disponível dos pesquisadores envolvidos e sendo transmitida por meio online.

Sobre as implementações nos patches, está sendo trabalhado a possibilidade do acoplamento de mais dispositivos a fim de criar um instrumento que possua uma massa sonora mais encorpada. Há também a pretensão de tornar o instrumento apto a criação de novas peças, desenvolvendo sua própria linguagem em uma pesquisa artística, também com a implementação de uma interface amigável e intuitiva, para que os usuários finais não precisem de ter um contato prévio com o pure data, por exemplo.

Atualmente o instrumento ainda funciona com um computador exercendo o papel de maestro, devido aos fatores supracitados sobre o isolamento social, mas o ideal é que esta função seja exercida por outro smartphone, com um patch finalizado onde os aparelhos escalados possam ocupar qualquer lugar na configuração do instrumento, que será configurada para as necessidades de cada proposta artística a ser construída.

5. Conclusão

Apresentamos neste trabalho a criação de uma orquestra de dispositivos móveis da Orquidea[5, 6]. Neste trabalho abordamos tanto a base tecnológica que sustenta nosso trabalho quanto o desenvolvimento de amplificadores cerâmicos para os dispositivos de nossa orquestra. Envolvendo áreas como artes (cerâmica), computação e música, esse trabalho buscou criar um ponto onde todas essas áreas pudessem dialogar a fim de construir algo em conjunto. Foi partindo deste lugar inter/transdisciplinar de pesquisa que surgiu a ideia da criação de um instrumento analógico-digital, que trouxesse um elementos de áreas distintas que permitiram uma criação coletiva. Partindo de uma metalinguagem onde temos uma mini orquestra de celulares, maestrados por um computador, nasce o con-

ceito do instrumento de cerâmica e silício. Nosso ferramental é ainda pautado sobre uma estrutura de rede e traz em si a discussão da sincronização e do controle de uma orquestra digital.

Este trabalho serviu ainda como experimento para discussões sobre arte digital, arte colaborativa e trabalho conjunto em criações artísticas, contribuindo para a pesquisa maior do laboratório pelo qual ela foi desenvolvida e ainda sendo um material para qualquer um que tenha interesse em discussões parecidas. Parte do resultado desta pesquisa pode ser visto no nosso canal do Youtube¹.

6. Agradecimentos

Este projeto encontra-se no contexto do Laboratório de Pesquisa ALICE (Arts Lab in Interfaces, Computers and Else) do Departamento de Computação e da Orquídea - Orquestra de Ideias. Agradecemos à Pró Reitoria de Extensão da UFSJ, por financiar esta pesquisa e também, a todos os membros do ALICE, que contribuíram para a evolução desse trabalho. Agradecemos também a PROPE / PROEX / PROAE e CNPq por financiar esta pesquisa.

References

- [1] Joseph Malloch and Marcelo M. Wanderley. The t-stick: From musical interface to musical instrument. In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, NIME '07, page 66–70, New York, NY, USA, 2007. Association for Computing Machinery.
- [2] Serge De Laubier. The meta-instrument. *Computer Music Journal*, 22(1):25–29, 1998.
- [3] Matthew Lee and David Wessel. Connectionist models for real-time control of synthesis and compositional algorithms. icmc, san jose. *International Computer Music Association*, 1992.
- [4] 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*, 48(4):316–330, 2019.
- [5] Flávio Luiz Schiavoni, João Teixeira Araújo, Igino de Oliveira Silva Junior, and Isabella de Melo Freitas. As lições aprendidas com a orchidea. *DEBATES - Cadernos do Programa de Pós-Graduação em Música*, 1(23):118–154, 2019.
- [6] 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.

¹<https://www.youtube.com/watch?v=c6nRrwjL2Y8>.

Rapid prototyping: using Wizard of Oz to emulate machine learning features for interactive artistic applications

Alessandra Aleluia^{1*}, Giordano Cabral¹

¹Mustic Research Group
Centro de Informática
Universidade Federal de Pernambuco (UFPE)

aaa6@cin.ufpe.br, grec@cin.ufpe.br

Abstract. In this paper, we present using the Wizard of Oz method to rapid prototyping machine learning features in interactive artistic applications. Machine learning systems often require time and resources until they were available to be executed. But in the era of agile movement where we have to test as soon as possible and in artistic solutions whose scope initially perhaps be unclear, there is a need for a way to fast testing hypothesis. We also briefly described Ama, a system for adult ballet training at home, which served as a proof of concept of our strategy.

1. Introduction

Machine learning is very often used in several applications, e.g. making predictions or to creating art. Although the design and implementation of systems that use it remain a contemporary challenge not only for developers but for designers too [1]. Machine learning classification techniques usually require a certain amount of data and time to train and tune the algorithms [2]. This scenario is considering they already had good datasets to use these techniques, which is not regularly the case.

Despite few datasets available for specif domains, artists so much as developers could not knowing in advance the final goal of the application and every step needed to get there. In recent years, with the ascending of agile movement and lean innovation, there's a need to rapidly test and evaluate solutions in the early stages of development to avoid wasting resources. It's expensive waiting for the full development of machine learning algorithms to get some feedback and we need to think in ways, with help of design techniques, to overcome these issues, for example, with participatory design [3]. Although there's no consensus about the prototyping method for machine learning experiences [4], in the case of interactive artistic applications we are investigating the use of the Wizard of Oz strategy to emulate machine learning features.

This paper is organized as follows: In Section 2 we discussed the ascending use of machine learning in artistic solutions, the issues around the use, the culture of lean innovation, and the Wizard of Oz method. Section 3 presents where we applied our approach, Section 4 was to describes it and in Section 5 we discuss what we learned through the process. We end this paper by outlining some limitations of the proposed approach and the hopes for the future.

*Supported by CNPq.

2. Background

Artists could benefit from the use of machine learning in their works using it in innovative ways. Now we have commercial software in the field of music AI to help in producing music without the need for music technologists to code. However a survey showed these people were a little skeptical about the current state of music AI but positive looking forward to the future of this field and many of them had built their own solutions using Python, SuperCollider, and Max/MSP [5]. Immersive experiences in mixed-media installations like *Where Is The Quiet?* [6] use machine learning to motivate individuals to relax, evoking meditative states of consciousness, and this kind of environment could support humans and provide new ways to appreciate art. Machine learning techniques are used in pose and gesture recognition, e.g., to support ballet training [7].

Machine learning algorithms improve through experience [8]. In general, they follow a well-established procedure: given a dataset, features were selected and extracted, these inputs were sent to the algorithm to train, some adjustments were made and the algorithm was ready to be used. It doesn't matter if it is a supervised, semi-supervised or non-supervised algorithm, these steps are almost the same as illustrated in Figure 1.

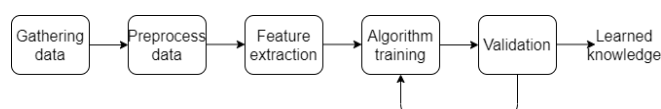


Figure 1: Machine learning process

In machine learning, it's fundamental to have a dataset to train your algorithms. Although we have some datasets available like ImageNet [9] for the computer vision field and efforts to develop platforms of openly available audio datasets to music computing like Freesound [10], they are not domain-specific and don't work for every application. Even though there are a few efforts to create APIs to support interactive machine learning, it doesn't solve the problem of not having time to create a very robust dataset to train machine learning algorithms. Research to explore the machine learning API design facilitating the usage for every kind of user didn't point out the problem of creating the dataset in a rapid prototyping approach [11].

Feature extraction is not a new problem and have been investigating for several years [12] and is complex

and time-demanding [13], [14], [15]. We have different types of features, to know - low, mid, and high-level features. In Camurri et al.'s work, we can see the mapping of low and mid-level features into sonification and they propose other high-level features to enhance the expressiveness [16].

However not every artist knows how to code and even she/he does, machine learning algorithms are very often time-consuming in terms of training classifiers and creating datasets. A priori, it's common to the artist doesn't have a clear idea of what she is going to do, it is an exploratory approach. Buxton (1997) [17] states that the artistic spec on design level is the hardest and the most important one, this implies a need for rapid prototyping to not waste an amount of time on something already difficult. The needs of lean innovation approach are required in this era of Industry 4.0 with artificial intelligence technologies and connected devices and it will remain relevant for the next era, Industry 5.0, of close cooperation between humans and machines with cognitive computing [18]. There are some clues that the Wizard of Oz might be a good method for prototyping machine learning experiences, however, it was not tested with users yet [19].

Wizard of Oz consists of a low-fidelity prototyping method where the human, known by the wizard, simulates the software's response and it is used in prototyping AI systems too [20]. In this method, the users believe that they acting with a real working system, but it's only a human controlling it. Back in time, the use of Wizard of Oz simulations is well known in human-computer interaction in natural language. It was not easy to acquire high-quality empirical data in some cases, then the Wizard of Oz is a way of dealing with it. One of the uses of this strategy is to overcome the computer's limitations. Humans are not machines so they might have issues regarding timing and consistency. Therefore, to make this difference small as possible environments could be built for conducting experiments [21].

3. Ama: adult ballet training at home

We proposed *Ama: treinos feitos para você* to help ballet dancers to train at home without a teacher. The goal is not to replace the teacher, Ama's main objective is to assist adult students in practicing out of the dance studio. Taking into account their own needs, providing automatic feedback.

The process to create Ama was very interactive and we made use of participatory design[20], always talking with ballet dancers, designers, and programmers. More than 25 dancers participated in the process and it is still a work in progress. We needed to understand our target audience, integrated them into the design process, and figure out how to build a good solution together without wasting much time on technical issues. We did not know all the requisites that our system requires, we can only perceive it when we test it, show it to the users, and receive their feedback - emphasizing the need for a rapid prototyping approach.

Ama is a very complex system that requires many steps until it's fully developed and ready to be used. At first glance, we mapped seven main modules, to be said: tracking, recognition, recommendation of ballet steps, calibration, evaluation, feedback, and demonstration. Every module has its own challenges to be implemented and tested. Tracking and recognition, as a major important part, could be simulated by our proposed approach and we could be further in our research without spending time creating the dataset and training the machine learning to recognize the skeleton and the joints status.

The prototype of our system, using Wizard of Oz to emulate the machine learning features and an expert system to provide automatic real-time verbal feedback, was tested by 19 adult ballet dancers. They had two sessions one week apart and each person performed five ballet steps each week. Participants were asked to answer a questionnaire with five questions on the Likert scale to evaluate the received feedbacks and after that, they were interviewed. The interviews were transcribed and analyzed later. The results indicate the system's feedback was relevant (considering aspects like information, reinforcement, motivation, analysis, and critical thinking which are the functions of feedback [22]) and participants are excited to use it in real life.

4. Wizard of Oz applied in feature extraction

As a result of time restrictions and not knowing the full ultimate goal of the experiment in advance, being an exploratory work, the need for rapid testing our hypothesis, as well as a proof of concept, made us use the Wizard of Oz's approach to emulate the machine learning features which will be required to our system. Our case is the adult ballet scenario that hasn't, in our prior knowledge, no specific domain dataset.

As we stated earlier, it's time-consuming to create a specific domain dataset from scratch. Due to the coronavirus pandemic (COVID-19) in 2020, Brazilian studios of dance were closed and we couldn't have easy access to the ballet dancers - which made it even difficult to create a dataset. In gesture recognition the goal is, given a person often represented by a skeleton, to identify the action made. We were inspired by skeleton's representation made by Kinect [23] and Android ML Kit¹.

The Wizard of Oz strategy is commonly applied to simulate the whole system by a person. But, in our case, the use of this technique was only a part of the full prototype as we can see in Figure 2. Applying the Wizard of Oz technique is an often approach in design, especially in the conception of the prototype. However, here we are suggesting the use of a very technical part usually designated to programmers or people with consolidated coding skills.

In order to provide relevant real-time feedback, specific features needed to be extracted when the ballet

¹<https://developers.google.com/ml-kit/vision/pose-detection>

dancer was performing. We developed an expert system based on rules to provide corrective and value verbal feedback. Features extracted using Wizard of Oz were the input of this system which gives the automatic feedback. Using this approach we can, with much less effort, rapidly prototype, test our hypothesis, and validate the system.

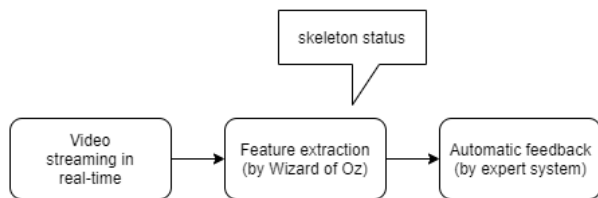


Figure 2: Our use of Wizard of Oz

Our wizard was responsible to enable or disable the checkbox and click on the correct radio according to the execution of the step demonstrated. The checkboxes were: right step, right side, right arms, right head, stretched foot. The radios were: En dehors (en dedans, little open, open, very open), Posture (aligned, misaligned), Leg's extent (not applicable, low, mid, high), Elbows (not applicable, low, right, high) and Shoulders (tense, relax). They represented the features that we were willing machine learning extracts.

The application was in real-time, the wizard must send it many times to reflect the changes made by the dancer over the time of performance. Every feature was pre-defined and the evaluation was taught to the wizard so she could judge based on the same parameters all the executions. In Figure 3, there is the interface manipulated by the wizard while she is seeing the dancer's performance. Of course, because the feature extraction relies on the human operating the machine, feedback timing is influenced by the wizard's skills sending the status to the system.

5. Discussion

The use of machine learning is increasing in computing and in the intersection field of art and technology too. But we can put more effort to make a dialogue between the technical side and the design side, which is so important in experimental artistic solutions. Here we are proposing the use of a very useful design technique - the Wizard of Oz -, an old friend of us in the field of human-computer interaction to discover new horizons in the field of machine learning. However, we are not simulating the whole system as we have seen before, but the technique is playing an important role as a part of a major process.

Indeed in our case, the machine learning features that we are emulating were based on the current knowledge about human skeleton recognition and tracking. We did not mean to propose something far away from the reality or the current state of the art - which would not necessarily mean a problem if it were. So we spent a little bit of time thinking and organizing what informations we wanted to extract

Figure 3: Interface manipulated by human to emulate the feature extraction

from the skeleton and formalizing them to be understandable for the wizard.

The system, an expert system based on rules, did not take too long to be implemented. We used Python language with the help of PySimpleGUI² to create the interface operated by the wizard. Most of the current machine learning applications are implemented in python, which is why we choose it - to make it easier in the future to integrate our solution with a real machine learning algorithm to extract the skeleton features. Our goal was to test our hypotheses - mapping some skeleton features to give useful real-time verbal feedback - and this approach was the way to rapidly prototype the system without the creation and training of a domain-specific dataset.

Time restriction and the difficulty to create a dataset from scratch were critical aspects that lead us to choose this approach, but they were not the only ones. We can go further. We need to think of new ways to approximate artists - and even programmers - who do not know how to build a very robust and reliable machine learning algorithm. Show them that we have ways to collaborate, and in the very beginning, there is no need to worry about such technical issues. We can go beyond it and test and re-test our artistic solutions. Solutions with machine learning sometimes are seeing as a big technical challenge, and in fact, they are. But we can not let it stop us. We can overcome this challenge in the early steps to make us free to try

²<https://pysimplegui.readthedocs.io/en/latest/>

and test and make mistakes in the beginning.

6. Conclusion, limitations and future work

We presented a fast, easy and innovative way to prototype systems that made use of machine learning. This strategy could reduce the technology gap for people who do not want to waste much time implementing machine learning systems and focus more on the result instead of the technique. But, in terms of this work's limitations, maybe the person who emulates the machine learning features might be a little greedy and the feature extraction simulated were not in the granularity desired when it comes to real implementations - which is not necessarily a bad problem because it may push through the limits of the state of the art of machine learning techniques.

It's important to recall many questions are still open and many research opportunities are available. This paper only states the first step of a whole recognition system. We intend to implement machine learning algorithms that our Wizard of Oz strategy emulated. If we could track accurately human skeleton joints using an RGB camera, we will use it to train these algorithms and provide useful feedback. The correct tracking can be suitable for others problems that deal with gesture recognition too. We are looking forward to seeing other researchers in this diverse field making use of this method to overcome machine learning issues and making a lot of contributions in artistic areas.

References

- [1] Qian Yang, John Zimmerman, Aaron Steinfeld, and Anthony Tomic. Planning adaptive mobile experiences when wireframing. In *Proceedings of the 2016 ACM Conference on Designing Interactive Systems*, pages 565–576, 2016.
- [2] Aized Amin Soofi and Arshad Awan. Classification techniques in machine learning: applications and issues. *Journal of Basic and Applied Sciences*, 13:459–465, 2017.
- [3] Tone Bratteteig and Guri Verne. Does ai make pd obsolete? exploring challenges from artificial intelligence to participatory design. In *Proceedings of the 15th Participatory Design Conference: Short Papers, Situated Actions, Workshops and Tutorial-Volume 2*, pages 1–5, 2018.
- [4] Graham Dove, Kim Halskov, Jodi Forlizzi, and John Zimmerman. Ux design innovation: Challenges for working with machine learning as a design material. In *Proceedings of the 2017 chi conference on human factors in computing systems*, pages 278–288, 2017.
- [5] Shelly Knotts and Nick Collins. A survey on the uptake of music ai software. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 499–504, 2020.
- [6] McLean J Macionis and Ajay Kapur. Where is the quiet: Immersive experience design using the brain, mechatronics, and machine learning. In *NIME*, pages 335–338, 2019.
- [7] Margaux Fourie and Dustin van der Haar. Ballet pose recognition: A bag-of-words support vector machine model for the dance training environment. In *International Conference on Information Science and Applications*, pages 317–325. Springer, 2018.
- [8] Tom M Mitchell et al. Machine learning. 1997.
- [9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [10] Eduardo Fonseca, Jordi Pons Puig, Xavier Favory, Fred-eric Font Corbera, Dmitry Bogdanov, Andres Ferraro, Sergio Oramas, Alastair Porter, and Xavier Serra. Freesound datasets: a platform for the creation of open audio datasets. In *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93. International Society for Music Information Retrieval (ISMIR), 2017.*
- [11] Francisco Bernardo, Michael Zbyszynski, Mick Grierson, and Rebecca Fiebrink. Designing and evaluating the usability of a machine learning api for rapid prototyping music technology. *Frontiers in Artificial Intelligence*, 3(13):1–18, 2020.
- [12] D.M Gavrilu. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73(1):82–98, 1999.
- [13] Basura Fernando, Elisa Fromont, and Tinne Tuytelaars. Mining mid-level features for image classification. *International Journal of Computer Vision*, 108(3):186–203, 2014.
- [14] Vassileios Balntas, Andreas Doumanoglou, Caner Sahin, Juil Sock, Rigas Kouskouridas, and Tae-Kyun Kim. Pose guided rgbd feature learning for 3d object pose estimation. In *Proceedings of the IEEE international conference on computer vision*, pages 3856–3864, 2017.
- [15] Haikuan Wang, Feixiang Zhou, Wenju Zhou, and Ling Chen. Human pose recognition based on depth image multi-feature fusion. *Complexity*, 2018, 2018.
- [16] Antonio Camurri, Gualtiero Volpe, Stefano Piana, Maurizio Mancini, Radoslaw Niewiadomski, Nicola Ferrari, and Corrado Canepa. The dancer in the eye: towards a multi-layered computational framework of qualities in movement. In *Proceedings of the 3rd International Symposium on Movement and Computing*, pages 1–7, 2016.
- [17] Bill Buxton. Artists and the art of the luthier. *ACM SIGGRAPH Computer Graphics*, 31(1):10–11, 1997.
- [18] Banu Ozkeser. Lean innovation approach in industry 5.0. *The Eurasia Proceedings of Science Technology Engineering and Mathematics*, (2):422–428, 2018.
- [19] Jacob T Browne. Wizard of oz prototyping for machine learning experiences. In *Extended Abstracts of the 2019 CHI Conference on Human Factors in Computing Systems*, pages 1–6, 2019.
- [20] Yvonne Rogers, Helen Sharp, and Jenny Preece. *Interaction design: beyond human-computer interaction*. John Wiley & Sons, 2011.
- [21] Nils Dahlbäck, Arne Jönsson, and Lars Ahrenberg. Wizard of oz studies—why and how. *Knowledge-based systems*, 6(4):258–266, 1993.
- [22] Elizabeth Gibbons. *Teaching dance: The spectrum of styles*. AuthorHouse, 2007.
- [23] Pichao Wang, Wanqing Li, Philip Ogunbona, Zhimin Gao, and Hanling Zhang. Mining mid-level features for action recognition based on effective skeleton representation. In *2014 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8. IEEE, 2014.

Expressividade de instrumentos musicais digitais - Just push play

Gabriel Lopes Rocha¹, Flávio Luiz Schiavoni¹

¹ ALICE - Arts Lab in Interfaces, Computers, and Everything Else

Departamento de Ciência da Computação

Universidade Federal de São João Del Rei

gbr.cdc@gmail.com, fls@ufs.br

Resumo. A criação de um instrumento musical digital é uma atividade que atrai músicos e pesquisadores há décadas e certamente possui problemas em aberto sobre os quais estas pesquisas tem se debruçado. Certamente, criar um instrumento pode ser uma tarefa simples que consiste em organizar uma interface de forma que ela sirva para fazer som. No entanto, esta interface simples pode estar distante do que consideramos um instrumento musical quando pensamos que gerar som pode não ser a única coisa que buscamos ao tocar um instrumento.

1. Introdução

Tocar um instrumento pode significar inicialmente utilizar um objeto para fazer som e isso nos traz muitas possibilidades para pensar este conceito, especialmente pelo fato de que este objeto pode ser, em princípio, qualquer coisa, e que este som também pode ser qualquer som. Quando pensamos em criar um instrumento musical digital (IMD), as possibilidades se tornam muito amplas, já que há diversos aparatos tecnológicos, sensores, ambientes, linguagens de programação, um universo de possibilidades que podem ser utilizados na construção criativa de um IMD. No entanto, este universo de possibilidades criativas e interessantes que pode auxiliar na construção de novos IMD não garantem que tal instrumento será criativo ou interessante ou que tocar este instrumento seja uma tarefa criativa ou interessante, independentemente do que os termos criativo e interessante possam significar.

Pensando sobre tocar instrumentos nos levou a pensar sobre esta ação de uma forma desconectada do criar instrumentos. Em português, o verbo tocar nos remete a um contato tátil, possivelmente com as mãos, o que traz em si a intimidade do toque que serviria de gesto para a produção do som. No entanto, ao pensar o instrumento musical como qualquer objeto, é possível que o toque não seja necessário para emitir seu som, já que câmeras, sensores de presença e sensores eletromagnéticos dispensariam este contato físico para gerar um estímulo. Este é o caso do Teremim, por exemplo, um instrumento eletromagnético que dispensa o toque para ser tocado.

É possível também notar que esta relação não está colocada desta forma em outras culturas. Encontramos no inglês o verbo “play”, que significa tocar instrumentos mas também brincar ou jogar. Tocar um instrumento se torna assim o mesmo que jogar qualquer jogo, dispondo o instrumento musical ao universo lúdico dos jogos e de ações de entretenimento. Talvez não coincidentemente, este é o verbo que encontramos nos aparelhos musicais. Desde a invenção dos aparelhos musicais que tocar uma música é

apertar o botão de Play, remetendo este gesto ao fazer musical e criando uma analogia entre pressionar um botão e tocar um instrumento. Mas será que podemos considerar que pressionar o botão de play é tocar um instrumento?

Iremos adotar em nossa discussão um botão de play em especial, como o de um tocador de MP3, que aqui chamaremos simplesmente de **botão**. Faremos isso para assumir que este é um dispositivo físico simples e limitado, que possui apenas este único botão de play. Iremos ignorar, com isso, os demais botões que podem existir em dispositivos deste tipo como volume, pause, stop, shuffle ou qualquer outra possibilidade que certamente tornam nosso instrumento muito mais interessante e complexo. Assim, ainda que isso possa parecer totalmente absurdo, iremos discutir este botão como um instrumento musical.

Antes de prosseguir com isso, é preciso pensar um pouco mais sobre nosso instrumento. Iremos assumir também que este nosso botão está conectado a algum sistema de som, como uma caixa de som bluetooth, para a emissão de seu som, mas que não iremos abordar este sistema na nossa discussão. Assumiremos ainda que este botão pode ser programado pelo músico / luthier em algum momento antes de sua apresentação por meio de alguma outra interface, como a USB, mas que isso também não será abordado aqui. Normalmente, a criação de um IMD se dá em 3 partes bem claras: uma interface para capturar gestos, um sintetizador para fazer o som e um mapeamento que conecta de alguma forma a primeira parte com a segunda. Adotaremos esta divisão tripartite para iniciar esta discussão.

2. A Interface

A primeira consideração que assumimos ao propor nosso botão como um instrumento musical foi de que ele seria limitado a apenas um único botão. Zappi e McPherson discutem a restrição como parte importante no design de novos instrumentos [1]. No entanto, tal limitação não impede que este botão seja tocado de várias formas e, sim, há várias formas de se tocar um único botão. Podemos apenas pressionar o botão uma vez, ou pressionar duas vezes, como o duplo clique do mouse, ou pressionar e manter pressionado por determinados períodos de tempo ou separar os eventos pressionar o botão e soltar o botão. Também podemos pensar que nosso botão pode ser programado como uma máquina de estado e que o botão pode ter funções diferentes dependendo do estado em que o instrumento se encontra. Assim, manter pressionado pode colocar o botão no modo tocar, no modo programar ou no modo repetir e em cada um destes estados, pressionar o botão ou soltar o botão pode ter uma funcionalidade diferente.

Ampliar as possibilidades de controlar o botão, com um repertório de gestos e combinações e máquinas de estados, pode trazer várias coisas interessantes para nosso instrumento como a) permitir mapear diferentes comportamentos para as diferentes possibilidades sonoras, b) permitir que ele seja explorado em busca destas possibilidades, e c) permitir que alguém se torne perito em tocar este instrumento e que se torne necessário praticar para tocar o instrumento com precisão.

Apesar de tudo isso ser possível, para manter nosso botão fiel ao proposto inicialmente, vamos assumir que a interface é um botão que ao ser pressionado toca um som, como em um CD Player ou MP3 Player.

3. O Sintetizador

Independentemente de como iremos definir as interações do músico com o botão, podemos assumir que a limitação da nossa interface não implica que o botão pode fazer mais ou menos sons. Na verdade, este único botão pode ser utilizado para tocar várias horas de música, como toda a discografia de um compositor em suas diversas interpretações, diferentes arranjos e em uma determinada ordem; acionar um sistema de síntese complexo com parâmetros que podem ser modificados pelas combinações de toques; tocar um sample de alguns poucos segundos; ou acionar um único oscilador de frequência fixa e envelope AR rápido com tempo de release pré programado.

Apesar de tudo isso ser possível, para manter nosso botão mais fiel ao proposto inicialmente, assumiremos novamente que ele se assemelha ao botão de um CD Player e que, por esta razão, irá apenas tocar músicas pré-programadas. Com isso, nosso botão pode tocar qualquer música, de qualquer gênero ou estilo, bastando para isso que ele tenha sido alimentado preliminarmente com estas músicas. Iremos incluir aqui mais uma condição de nosso botão e assumiremos que isso precisa ser feito antes de tocá-lo e que não daria para tocar alguma música que o mesmo não tenha sido programado em sua playlist.

A construção desta lista permite que façamos um paralelo entre as músicas que nosso botão é capaz de tocar (quando programado para isso) e o repertório de um intérprete ou grupo. A construção de um repertório é certamente um exercício obrigatório para qualquer instrumentista e também para compositores e isso não é considerado uma limitação para um instrumento musical. A construção de repertório também precisa ser feita antes e isso exige muito estudo, conhecimento musical e prática. Neste ponto, nosso botão não se difere de outros instrumentos se assumirmos como parte do tocar a construção do repertório, adequação de estilo, escolha de temas e de sua ordem.

Músicos ensaiam antes de se apresentarem, se tornam peritos em determinados estilos, se aperfeiçoam em determinados repertórios como romântico ou barroco. Talvez nosso botão seja menos limitado do que qualquer instrumento acústico já que ele poderia atuar em qualquer repertório e não depende de ensaios para ser tocado. No entanto, isso não significa que não seja necessário construir um repertório antes de tocar o botão e que isso implica em

muito estudo e conhecimento do repertório. O botão em si não traz limitações para ser tocado mas certamente cada tocador de botão pode ter um repertório diferente. Assim, imaginando dois músicos tocadores de botão, os botãozistas, podemos pensar que ambos manipulam o mesmo instrumento mas que o repertório de ambos é diferente e por isso cada um deles pode agradar um público diferente e ser considerado melhor ou pior botãozista que o outro.

4. O Mapeamento

Ao permitir diferentes comportamentos elencamos que deve ser necessário ter mais de um mapeamento em nosso botão. A escolha de mapeamento tem grande impacto no resultado final de um IMD[2, 3]. Ao pensarmos o IMD como uma estrutura tripartite formada por interface, sintetizador e mapeamento, escolhas e alterações feitas em qualquer uma dessas partes são capazes de mudar completamente a identidade do instrumento. A grande quantidade de possibilidades em cada uma destas partes é o que torna o desenvolvimento de IMDs tão interessante e desafiador. Para o nosso instrumento, mesmo com uma interface tão simples, ainda podemos capturar um conjunto gestos como pressionar, soltar, pressionar repetidamente ou segurar o botão. Cada uma destas ações pode ser mapeada para manipular um conjunto de parâmetros sonoros e isto pode ser combinado com um controle sensível ao contexto, similar ao que ocorre em videogames onde o mesmo botão executa ações diferentes dependendo da situação. Um botão pode fazer o personagem golpear e também interagir com um objeto que esteja próximo, dependendo do contexto. Com isso conseguimos expandir as possibilidades mesmo através de uma interface tão simples.

É interessante pensar que a proposta deste instrumento é trabalhar sobre um fluxo sonoro predeterminado e não com eventos isolados que podem gerar som. Ao tocar um instrumento acústico, o som é gerado pela interação com um objeto e isso pode significar eventos individuais que irão produzir determinados sons. Quando sopramos uma flauta com a embocadura correta ela produz um determinado som e existe uma relação entre o gesto e o som criado. Diferente disso, no caso do botão, ele irá trabalhar sobre uma música que já está pronta e previamente gravada, o que leva a pensar no nível de automação que determinado gesto pode disparar.

Parâmetros sonoros do sintetizador podem, por exemplo, ser organizados em diversas camadas sendo que, em mais baixo nível temos o controle nota a nota comum aos instrumentos acústicos e, a partir disso, podemos passar para camadas superiores, com ações que afetam frases e sessões da música. Podemos dar comandos que afetem instrumentos específicos ou todo o conjunto e pensar em parâmetros abstratos como tocar de maneira mais calma ou agitada. Neste contexto, aumentar o nível de automação em um IMD aumenta a capacidade de fazer coisas mais complexas com gestos simples mas tira a possibilidade do instrumentista executar ajustes mais específicos [4]. Sem nenhum grau de automação seria impossível desenvolver um IMD que permitisse tocar todos os instrumentos de

uma peça simultaneamente. Isto só se torna viável se partirmos do paradigma que o performer se comportará como o maestro de uma orquestra e os instrumentistas serão simulados pelo computador, de maneira automática. No entanto, uma automação total pode se tornar o nosso botão de play, que inicia um determinado fluxo de som e depois disso não há mais controle que o executante possa utilizar para alterar este fluxo.

Desta maneira podemos, a partir de uma interface simples, desenvolver um IMD que permita o controle sonoro de maneira complexa, como é o caso de nosso botão. Em qualquer IMD é necessário que o controle da interface ocorra de maneira intuitiva. Para isso pode ser preciso um certo nível de treinamento para desenvolver a familiaridade necessária com o instrumento. Após um tempo de exploração e prática, a ação de tocar pode ser adquirida de maneira natural, sem que o músico tenha que raciocinar ativamente sobre ela. Um bom esquema de mapeamento deve ser pensado para permitir que a interação com a interface aconteça de maneira exploratória[5], o que nos leva a ideia de “brincar” (play) com o instrumento. Combinar essa noção com a possibilidade de um controle assistido por algoritmos trás um novo horizonte de possibilidades para um IMD. Com isso, é possível tocar uma música inteira usando apenas um botão e ainda fazer isso de maneira interessante de forma a assumir que as possíveis limitações deste instrumento não se encontra na interface botão, mas nos mapeamentos e automações que o instrumento pode ou não possuir.

5. A Idiomática

Há instrumentos polifônicos que são capazes de fazer acordes e há instrumentos monofônicos que só emitem um único som de cada vez. Esta capacidade ou não de produzir mais de uma nota musical pode, muitas vezes, ser considerada uma limitação do instrumento e as limitações de um instrumento estão associadas a sua idiomática. O termo idiomático tem origem na palavra idioma. Segundo o dicionário, idioma trata-se da língua falada por um povo, com todos seus elementos fonéticos e gramaticais. Idiomático, por sua vez, pode ser entendido como aquilo que é característico de um idioma. Um exemplo são expressões como “quebrar a cara” cujo significado é diferente do significado literal das palavras que a compõem. Estas expressões são características de um determinado grupo que compartilham um entendimento do sentido verdadeiro de palavras e que conseguem compreender a organização das palavras para além dos significados individuais de cada palavra em uma sentença.

Podemos a partir disso expandir estes significados para o idioma e idiomática de um instrumento musical. Idioma neste contexto seria o conjunto de elementos e possibilidades que são características do instrumento como o timbre, articulação, dinâmica entre outros. Idiomática é a forma como tais elementos podem ser apresentados e explorados pelo instrumentista[6, 7]. Certos gestos e métodos ao tocar determinado instrumento são incentivados pela forma e afinação do instrumento musical em si. Para além disso, podemos refletir sobre a idiomática de determinado grupo, movimento cultural ou período histórico.

Assim, é possível entender, por exemplo, que há diferenças idiomáticas entre o violão para repertório clássico e o violão para o repertório de blues.

E isso pode existir para nosso botão. A idiomática de nosso botão pode ser diretamente influenciada pelo repertório selecionado e as formas de controle implementadas. Deixar um trecho em loop, aplicar um filtro passa banda ou remover totalmente uma faixa de frequência pode ser muito comum para um repertório de música techno mas pouco provável para trechos musicais clássicos, por exemplo. O método de interação com um IMD é bastante sensível a decisões tomadas na camada de mapeamento e isso é influenciado pela idiomática escolhida para este instrumento. Em nossa experiência, este grande universo de escolhas na programação de um IMD é tão amplo que pode ser desorientador[8].

Existe o risco de o desenvolvedor de IMDs se perder num ciclo vicioso de experimentação e nunca chegar a um resultado concreto. Uma maneira de lidar com isso é limitar as nossas escolhas para um subconjunto mais específico, definindo com isso, a idiomática do instrumento em seu processo de criação. Podemos pensar na criação de instrumento a partir de uma idiomática desejada, como um conjunto do que queremos de forma a limitar o universo infinito de possibilidades. Tal idiomática pode se basear em instrumentos ou sistemas já existentes e trazer limitações simples, como a característica de ser monofônico. A partir disto entramos para a fase de implementação com um objetivo em mente e isso permite tratar as características como virtudes e não como limitações do instrumento criado.

6. Performance e Expressividade

Instrumentos musicais digitais devem buscar ser expressivo. Vemos esse assunto ser abordado em diversos trabalhos sobre luthieria digital. Para responder o porque disso, podemos perguntar por que tocamos instrumentos musicais. É fácil perceber que nessa atividade o processo de tocar é tão importante quanto o resultado sonoro. O botão, como estamos propondo, é bastante eficiente na tarefa de tocar músicas, principalmente se considerarmos o caso mais simples em que, para isto, basta pressioná-lo uma vez.

Apesar de ser eficiente em sua tarefa de tocar música, nosso botão pode falhar no quesito performático caso a proposta artística do mesmo seja alcançar um público que não está interessado em assistir um concerto onde o músico apenas aperta um botão e assiste a música acontecer. Normalmente, esperamos que uma performance musical traga consigo a perícia no tocar e não apenas uma boa escolha de repertório. Por outro lado, se os instrumentistas em um concerto seguem uma escolha de repertório com instruções descritas em uma partitura que diz exatamente o que devem tocar, estaríamos interessados apenas nas exibições de habilidade com os instrumentos? Neste contexto, expressividade pode ser entendida em dois momentos diferentes. O compositor ao criar a sua obra canaliza nestas intenções, significados e sentimentos. A capacidade de comunicar estes elementos é característica da música em si [9]. O instrumentista, ao interpretar uma obra, irá dar novas nuances a

estes elementos de forma a tornar a sua performance única. Esta ação se dá através da aplicação de técnicas vindas de sua habilidade em tocar o instrumento.

Estas técnicas, individuais e subjetivas, atendem ao conceito de ações performáticas, que pode ser definido como “estratégias e mudanças que não são marcadas em uma partitura” [4]. Tais ações performáticas podem ocorrer em diversos níveis da organização estrutural da música já que instrumentistas tendem a usar variações para destacar elementos da estrutura, como frases e seções. Uma peça pode também ser tocada de maneira a destacar um determinado humor, como triste, animado ou solene. O entendimento de conceitos complexos como sentimentos depende de um vocabulário compartilhado entre o intérprete e os ouvintes.

Poepel[10] propõe que músicos transmitem a sua expressão através de um sistema de codificação e que essa codificação ocorre através de marcações na performance de maneira similar à qual definimos ações performáticas. Cabe então ao público decodificar estas marcações para compreender a intenção do instrumentista. Estes conceitos se relacionam bem com as noções de repertório e de idiomática. Músicos que compartilham um repertório também irão compartilhar a sua idiomática, que servirá de meio para a codificação da expressividade e determinará as ações da performance musical.

Se pensarmos na organização de um IMD, o sintetizador nos fornece os parâmetros sonoros e a interface os meios de interação mas permitir o controle do som por meio da interface não é suficiente para alcançar expressividade[11]. A expressividade vem da habilidade de um instrumentista experiente conseguir transparecer suas intenções através da técnica. Por essa razão, o instrumento deve ser complexo o suficiente para permitir que essa técnica se desenvolva através de treinamento trazendo outro conceito importante que é o da virtuosidade, o quanto o músico consegue extrair o máximo do instrumento através de sua habilidade treinada. Instrumentos acústicos evoluíram ao longo do tempo através de toda uma comunidade que explora suas possibilidades e tais comunidades trocam informações entre si levando ao desenvolvimento da idiomática e repertório do instrumento. A questão para um novo IMD é como alcançar expressividade sem ter que passar por essa fase de amadurecimento.

7. Conclusão

Este texto é um ensaio e traz divagações sobre a construção de um IMD partindo de uma ideia pouco convencional de tocar um único botão para fazer som. Estas divagações não pretendem discutir se isso é ou não tocar um instrumento mas ser um exercício de criação que pode nos levar a pensar o que pode ser tocar um instrumento, quais os níveis de automação e complexidade que podemos ter, como pensar performance, idiomática, repertório e outras questões tão importantes para a criação de IMD. Nosso botão tem uma entrada complexa e isso garante a saída complexa pois temos um nível enorme de automação que podem ser

pensadas como decisões que foram tomadas antes da performance e que, por isso, não trará uma performance expressiva ou singular. Há um esforço para converter entrada em saída e há muito esforço para codificar o que é a entrada e pouco em converter esta entrada em diferentes saídas. Complexidade do esforço pode ser algo interessante a se pensar na criação de um IMD.

Pensar na expressividade deste instrumento nos leva ainda a pensar que não seria possível improvisar com o mesmo. A diferença entre compor e improvisar está na necessidade de aceitar o tempo real como parte fundamental da improvisação, o que não é fundamental para a composição. Compor permite reorganizar os eventos no tempo em qualquer hora pois o tempo da composição não significa o tempo real. E o tempo real parece ser algo fundamental para a performance musical, que talvez precise de um pouco mais do que apertar um botão de play.

Os autores gostariam de agradecer os membros do Laboratório ALICE (Arts Lab in Interfaces, Computers, and Everything Else) da Universidade Federal de São João del-Rei e o apoio financeiro institucional do CNPq, FAPEMIG e da PROPE/PROAE/UFSJ.

Referências

- [1] Victor Zappi and Andrew P McPherson. Dimensionality and appropriation in digital musical instrument design. In *NIME*, volume 14, pages 455–460. Citeseer, 2014.
- [2] Andy Hunt, Marcelo M Wanderley, and Ross Kirk. Towards a model for instrumental mapping in expert musical interaction. In *ICMC*. Citeseer, 2000.
- [3] 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.
- [4] Alexis Kirke and Eduardo R Miranda. *Guide to computing for expressive music performance*. Springer Science & Business Media, 2012.
- [5] Andy Hunt and Ross Kirk. Mapping strategies for musical performance. *Trends in gestural control of music*, 21(2000):231–258, 2000.
- [6] Elias de Oliveira Ferreira. *A obra musical Repentes de Pedro Cameron: Influências idiomáticas do violão como recurso composicional*. PhD thesis, 2016.
- [7] Sergio Antonio Caldana Battistuzzo et al. Francisco Araújo: o uso do idiomatismo na composição de obras para violão solo. 2009.
- [8] Rômulo Vieira, Gabriel Rocha, and Flávio Schiavoni. Current research on the use of hci in decision-making to build digital musical instruments: A survey. In *Proceedings of the 19th Brazilian Symposium on Human Factors in Computing Systems, IHC '20*. Association for Computing Machinery, 2020.
- [9] Josep Lluís Arcos, Ramon López De Mántaras, and Xavier Serra. Saxex: A case-based reasoning system for generating expressive musical performances. *Journal of New Music Research*, 27(3):194–210, 1998.
- [10] Cornelius Poepel. On interface expressivity: A player based study. In *NIME*, volume 5, pages 228–231. Citeseer, 2005.
- [11] Christopher Dobrian and Daniel Koppelman. The’e’in nime: Musical expression with new computer interfaces. In *NIME*, volume 6, pages 277–282, 2006.

Desafios da Internet das Coisas Musicais

Alexandre Silva de Almeida^{1,2}, Rômulo Vieira^{1,2},
Rafael Sachetto Oliveira², Flávio Luiz Schiavoni^{1,2}

¹ ALICE - Arts Lab in Interfaces, Computers, and Everything Else

²PPGCC - Programa de pós Graduação em Ciência da Computação

UFSJ - Universidade Federal de São João del-Rei

São João del-Rei, MG – Brasil

silvaale2@gmail.com, romulo_vieira96@yahoo.com.br

sachetto@ufsj.edu.br, fls@ufsj.edu.br

Resumo. A Internet das Coisas Musicais é uma área de pesquisa que pretende levar a conectividade da Internet das Coisas para o campo da música e das artes. Junto a esta tecnologia surge a possibilidade de conexão de diferentes coisas musicais em um ambiente de concerto ou de criação artística que permitiria, por exemplo, a participação do público nestes processos, tanto de maneira presencial, por meio de uma rede local, quanto remoto, por meio da Internet. Neste trabalho trazemos algumas discussões sobre a IoMusT e também suas possibilidades e desafios.

1. Introdução

O termo Internet das Coisas ou do inglês *IoT - Internet of Things* foi popularizado em 1999, mas desde os anos 80 que o tema era abordado pela União Europeia [1]. Seguindo a descrição da IEEE, a Internet das Coisas pode ser definida como uma rede de itens, incorporada com sensores que podem ser conectados à Internet. Esta definição pode ser aprofundada ao abranger uma numerosa quantidade de objetos e dispositivos conectados diretamente um ao outro ou conectados pela rede, utilizando uma identificação única para cada elemento e operando de maneira inteligente (sem a interferência dos usuários), fazendo com que troquem e processem informações de qualquer lugar do mundo e em qualquer hora.

Para uma melhor percepção, a Internet das Coisas pode ser vista como um segmento da internet, conectando dispositivos (coisas) heterogêneos. As “coisas”, no que lhe concerne, são definidas como conjuntos de tecnologias e objetos em sistemas embarcados que possibilitam a comunicação, viabilizando a interação entre diversos elementos pela rede.

2. Internet das Coisas Musicais (IoMusT)

A Internet das Coisas Musicais ainda é um assunto desafiador e, com o avanço da Internet das Coisas, este cenário também tem ganhado atenção e tem gerado diversas discussões [2]. Este campo é descrito como “conjunto de interfaces, protocolos e representações de informações relacionadas à música que permitem serviços e aplicativos que atendem a um propósito musical com base nas interações entre seres humanos e coisas musicais ou entre as próprias coisas musicais, no domínio físico e/ou digital” [3]. Neste contexto, uma “coisa” pode ser definida como “um dispositivo de computação capaz de detectar, adquirir, processar

ou atuar e trocar dados que atendam a um propósito musical”.

De forma geral, a Internet das Coisas Musicais pode ser vista como um caso particular da Internet das Coisas aplicada ao contexto musical. Neste modelo, diversos componentes conectados podem interagir entre si, bem como pessoas com coisas musicais, objetos musicais com outros objetos musicais e etc. Além disso, artistas e membros da audiência podem participar através de ambientes locais e/ou remotos.

Possíveis cenários de emprego para Internet das Coisas Musicais passam pela gravação remota em estúdios, utilizando ferramentas de acompanhamento e integração com elementos presentes em outros locais de gravação, permitindo que esta ação seja compartilhada, aceitando inclusive que instrumentos e acessórios necessários para produção de conteúdo musical estejam distribuídos geograficamente [4]. Ainda, pode ser utilizada em mixagens de áudio ao vivo através da rede, possibilitando a participação em tempo real no espetáculo, e finalmente, proporcionar novas experiências para sessões de improviso (*jam sessions*), comportando instrumentos eletroacústicos e coisas musicais [4].

3. O Ecosistema da IoMusT

A infraestrutura da Internet das Coisas Musicais deve conseguir comportar um ecossistema heterogêneo de dispositivos, assim como a interação dos músicos com o público em tempo de real, a uma baixa latência, alta confiabilidade e excelente sincronismo, seja no ambiente local ou remoto [5, 6].

Como exemplo, cita-se a interação do público com elementos gráficos e sonoros, manipulando-os através de *smartphones*, o que gera um ambiente de realidade interativa. Existe também a possibilidade de abarcar a realidade virtual em 360°, onde membros da audiência podem ser posicionados no centro de um palco em um festival ou no meio de uma orquestra de música erudita [7, 8].

Para exemplificar melhor este ecossistema, podemos considerar sua integração em três grupos tecnológicos:

Coisas Musicais: na abordagem da IoMusT, crescem novas tendências relacionadas a dispositivos musicais inteligentes com sensores e atuadores capazes de conectar-se a

Internet e que poderão impactar a relação entre espectadores e artistas/músicos, alcançando os desafios como sincronismo e interoperabilidade. As coisas musicais podem ser usadas para reprodução de conteúdo musical, aprendizagem de experiências relacionadas a música e controle dos demais objetos conectados à rede. Elas são exemplificadas na figura de instrumentos ou dispositivos inteligentes, ou ainda, qualquer outro elemento que esteja na rede e que seja utilizado para controlar, gerar, rastrear ou processar conteúdo musical.

Conectividade: através da infraestrutura de redes locais e remotas, as coisas musicais mantêm sua comunicação, seja em rede sem fio ou cabeada, através dos recursos de *hardware* e *software* disponíveis neste tipo de ambiente, bem como as tecnologias e protocolos de comunicação. A comunicação é crucial na Internet das Coisas Musicais, tendo como requisitos importantes o sincronismo, conectividade em tempo real, comunicações com baixa latência, alta confiabilidade, segurança e qualidade, características primordiais, principalmente em apresentações com interações ao vivo, com uso de realidade virtual e/ou aumentada [3].

Aplicativos e serviços: os aplicativos auxiliam a interatividade entre os objetos e humanos, provendo diversos recursos e serviços para os usuários e músicos [3]. Quando aplicados à Internet das Coisas Musicais, eles geram serviços com uma direção totalmente ampliada, possibilitando o alcance de cenários de grandes escalas, como, por exemplo, participações em ambientes *online* com *interfaces* virtuais, possibilitando que centenas ou milhares de participantes sintam proximidade uns com os outros, como se estivessem participando *in loco* do evento. Outro serviço plausível de ocorrer é a partilha de recursos presentes em objetos e em instrumentos musicais inteligentes, bem como o controle sobre a iluminação, sugestões de conteúdo ou ritmo musical, que com a coleta e análise das informações, novas experiências podem ser criadas ou melhoradas [3].

4. As possibilidades da IoMusT

Algumas possibilidades de interações podem ser proporcionadas pela Internet das Coisas Musicais, por meio dos diversos cenários participativos co-localizados em uma sala, evento ou *show*, ou ainda em cenários abertos reunindo diversos participantes em um ambiente totalmente virtual. Quando combinado a outras tecnologias, como a realidade virtual e realidade aumentada, o dinamismo de criação, experiência e aprendizagem musical se expande, permitindo ao público uma participação mais próxima, com a sensação de estar ao vivo no evento ou participando diretamente com os músicos na apresentação, ou concerto. Por meio dos recursos oferecido pela realidade aumentada, as performances interativas e participativas em concertos, por exemplo, podem ser aprimorados com experiências entre a participação da plateia e dos músicos em ambientes virtuais envolvendo dispositivos interativos.

A Internet das Coisas Musicas estende os benefícios, recursos e aperfeiçoamentos entre artistas e

músicos, equipamentos, dispositivos, e o comportamento do público. Também propõe melhorias e experiências entre os fornecedores de equipamentos musicais. Como exemplos, podemos imaginar a utilização de *smartphones* para controlar o som de instrumentos musicais inteligentes, ou ainda, formas de interagir através de relógios ou pulseiras inteligentes. O público, os músicos e os dispositivos podem interagir durante um *show*, de forma que os músicos tenham um *feedback* da plateia através de capturas fisiológicas, obtendo estímulos que podem controlar o timbre dos instrumentos ou comportamento dos equipamentos de palco, como iluminação, telões, entre outros. Da mesma forma, os sons de instrumentos inteligentes podem ser enviados através de vibrações em forma de ritmos para os vestíveis inteligentes do público [5].

5. Desafios da Internet das Coisas Musicais

Com o surgimento e crescimento da Internet surgiram muitas oportunidades envolvendo integração de músicas com redes, explorando domínios das apresentações musicais em redes (*Networked Music Performances* - NMPs), com criação de tecnologias de *hardware* e *software* como *codecs*, estruturas, protocolos de baixa latência para as apresentações musicais remotas, entre outras. No entanto, alguns desafios ainda estão presentes para tornar esta solução amplamente disponível. Eles são melhores debatidos a seguir.

Protocolos de comunicação: A infraestrutura tecnológica precisa de alguns avanços para conseguir prover comunicação entre os dispositivos e conectividade necessária para as coisas musicais. Isto engloba sensores capazes de capturar áudio e informações específicas de um emissor e um receptor que compreenda a leitura dessas informações conseguindo processá-las e respondê-las, permitindo a interação e sincronismo das coisas e dos envolvidos. Para isso, precisamos de uma rede com componentes e dispositivos que reaja com eficiência, baixa latência, confiabilidade e segurança.

Geralmente, a Internet das Coisas Musicais propõe criações e adaptações das camadas e protocolos da pilha TCP/IP. Podemos utilizar, como exemplo, para garantir as baixas latências na camada física, as mensagens serem essencialmente curtas podendo ocasionar restrições nas taxas de dados, mas aumentando o desempenho da comunicação. Na camada de rede, os protocolos de roteamentos precisarão ser otimizados para apresentarem menos latência, além de métodos de decisões de roteamento distribuído, capazes de minimizar o atraso da comunicação em rede. Pode ser necessário ainda novas técnicas de controle e gerenciamento baseados em otimização que deverão conseguir configurar e alocar os recursos adequados do plano de dados em diferentes domínios para criar serviços de ponta a ponta com baixa latência [5].

Nas abordagens de análise de dados devem ser implementados técnicas e ferramentas capazes de armazenar e processar grandes quantidades de dados relacionados a música, como *Big datas*, *web* semântica, ontologias, de

modo que, quando necessário e solicitada as informações, elas estejam disponíveis.

Quebras de paradigma: Algumas quebras de paradigmas devem ser exploradas, como: i) Meios com que os integrantes da plateia tenham interatividade diretamente com o artista/músico no palco; ii) Recursos de métodos de composição de músicas que proporcionam autonomia aos integrantes, de forma individual ou em conjunto, com diversos indivíduos, provendo sensações mútuas sem comprometer a qualidade e performance, visando a diversidade de cada indivíduo em sua percepção de sensibilidade, preferência e aptidão; iii) Mecanismos que possibilitam, criação e ensaios de músicas e *shows* de forma remota, utilizando a tecnologia para integrar cada expressão participativa, seja ela individual ou em grupos; iv) Questões relacionadas à segurança, privacidade das informações e desempenho [5].

Desempenho: O bom funcionamento da IoMusT é crucial para expansão de aplicativos e serviços que remodelará a forma de interação entre músicos em ambientes remotos. Para isso, é indispensável assegurar o desempenho de forma que alguns requisitos necessários para execução de áudios em redes sejam estabelecidos, como atraso de rede, qualidade de áudio, sincronização satisfatória entre os músicos, além da coleta e análise de experiências interativas. Assegurar captura e transmissão de fluxos de áudio pela Internet ainda é um desafio, pois a sincronização e o desempenho da rede podem ser comprometidas pelos atrasos de pacotes e atrasos de processamento com os músicos estando em diferentes locais geograficamente distribuídos.

Baixa latência: Mesmo com a conexão móvel 5G oferecendo altas taxas de transmissão de dados, a latência acaba sendo objeto de muitos estudos devido as rigorosas exigências que a Internet das Coisas Musicais apresenta em relação a este tópico, principalmente se tratando de transmissões ao vivo com interconexões de diversos instrumentos musicais.

Por outro lado, outros mercados estão estimulando a melhoria das taxas de transmissão em rede, como, por exemplo, classes emergentes de serviços, como jogos digitais, telepresença, realidade virtual, realidade aumentada e controle de missão crítica. Uma alternativa seria a utilização de estudos envolvendo “*Tactile Internet*” que operam com uso de latências bastante baixas e com atrasos de pouquíssimos milissegundos (não excedendo 5 ms), abrangendo recursos para tecnologias com frequências sem fio na faixa de 10 a 300 GHz com taxas de dados em Gb/s por segundo para distâncias curtas, tornando crucial a sua adoção nas coisas musicais [5].

Assim, baixa latência e atrasos como captura e processamento de sons, toques e notas de equipamentos musicais, transmissão das vozes dos músicos, *feedback* visual de movimentos e gestos de outros integrantes em tempo real se tornam cruciais para a Internet das Coisas Musicais. É necessário alcançar requisitos rigorosos de latências na casa de algumas dezenas de milissegundos, com limites de atraso estimados entre 20 e 30 ms, podendo

variar conforme o cenário e distância para tornar tal realidade possível [9].

Sincronização: A comunicação destas coisas ocorrerá, em sua maioria, em topologias remotas em locais distribuídos ou por meio da internet. De tal maneira, a sincronização acaba sendo mais um requisito fundamental a ser analisado para garantir seu desempenho satisfatório no fluxo de dados emitidos, principalmente pelos objetos e dispositivos que não compartilham o mesmo relógio. Algumas técnicas podem ser adotadas para alinhamento na transmissão dos fluxos de modo que o envio de dados de áudio, vídeo e controle possa ocorrer sem comprometer a qualidade, além do controle de perda de pacotes de rede ou para resolver o problema de tremulação de rede, evitando um pacote atingirem seu destino após o tempo de reprodução programada [9].

Para a Internet das Coisas Musicais, podem ser utilizados o controle com pacotes de áudios e/ou vídeos coesos com a data e hora, com os protocolos RTP/RTCP e NTP [10, 11]. Há algumas alternativas para sincronização dos relógios, de forma que os pacotes de controle com as informações do relógio principal possam ser anexados aos dados de *streaming* de áudio e enviados ao receptor para que os relógios sejam sincronizados. Nesta situação, enquanto a sessão estiver estabelecida, o NTP envia atualizações periódicas. Contudo, este protocolo possui atrasos consistentes em comunicações remotas, podendo comprometer a comunicação em tempo real.

A latência e a sincronização, no entanto, dependem do tipo de dado a ser compartilhado. Em comunicações que utilizam dados MIDI, por exemplo, é possível alcançar uma latência muito mais baixa que na troca de áudio [12] e é possível utilizar a sincronização de relógios MIDI, uma espécie de relógio global, onde um nó principal envia o relógio a todos os clientes e os valores são baseados na frequência de *clock* do *hardware* do nó principal [9]. Tais soluções, no entanto, precisam ser repensadas para comunicações baseadas em stream de áudio e vídeo.

Integração e facilidade de participação: Garantir a comunicação e interoperabilidade dos mais diversos dispositivos musicais também é um desafio, tendo em vista que não há compatibilidade total entre os instrumentos eletrônicos e não eletrônicos e entre as informações gestuais e sonoras. Alguns protocolos proprietários possibilitam essa comunicação, ao passo que outros equipamentos usam a *Interface* MIDI para transporte de mensagens [9]. No entanto, não é possível garantir que dispositivos terão compatibilidade de dados e algumas conversões poderão ser necessárias para garantir a interoperabilidade e integração.

6. Discussão

A IoMusT se mostra como uma área com grande potencial artístico e computacional, mas que certamente possui desafios tecnológicos que dificultam a sua implementação. Latência, sincronização, baixo consumo elétrico e outras questões se tornam parte dos obstáculos a serem ultrapassados para tornar esta tecnologia amplamente difundida e

utilizada. Pensando em como viabilizar esta tecnologia, surgem possibilidades simples que podem ser empregadas.

Uma possibilidade para criar os aparelhos que vão compor esta área é utilizar um equipamento de uso comum na IoT para a prática musical. Já existem sensores, câmeras e outros dispositivos que se conectam pela rede que poderiam ser explorados de maneiras musicais. Outra perspectiva é adaptar instrumentos, equipamentos de áudio (microfones, mesas de som, alto-falantes, etc.), unidades de efeito (pedais de guitarra e *plugins*) e ferramentas que auxiliam na apresentação (como canhões de luz) para serem usados com conexão à internet [13]. Para isso ocorrer, é desejável atribuir-lhes a capacidade de remoção ou inserção de botões de controle e sensores, de modo que os usuários possam adaptá-las para o novo ambiente que se vislumbra diante deles. Além das mudanças físicas, alterar o código de controle e o *software*, quando presentes, também pode permitir a adequação de um determinado dispositivo a um novo contexto. Ainda, algumas funcionalidades podem ser incorporadas para transformar seu uso em uma atividade coletiva.

É desejável utilizar esta tecnologia para permitir que o público tenha maior imersão nas apresentações musicais. Isso pode ser alcançado equipando-o com ferramentas como *smartphones*, etiquetas RFID e outros. Assim, a capacidade de metamorfose de elementos tradicionais para coisas musicais é ainda mais necessária, dado que o mesmo objeto estará em diferentes circunstâncias e desempenhando diversos papéis. Estes dispositivos, somados a uma capacidade de conexão por rede, munida de serviços e aplicações de cunho musical, criam um ambiente interoperável, responsável por conectar músicos e membros da audiência, o que multiplica a possibilidade de interação em espetáculos musicais.

Neste ponto, surgem outros desafios, que não são mais somente tecnológicos mas também artísticos, relacionando-se com tópicos como criatividade e possibilidade de criação artística em ambientes compartilhados [14]. É necessário pensar no resultado estético da utilização destes equipamentos, nas formas de criação artística coletivas, e na participação do público nestes ambientes. Tais questões, ultrapassam as questões tecnológicas e se colocam como outros desafios para esta emergente área.

7. Conclusão

A área de Internet of Musical Things proporciona novas perspectivas para a prática musical, mediando a interação de musicistas com seus pares ou de musicistas com membros da audiência através de diferentes recursos computacionais. Os concertos ao vivo, gravações em estúdio e aprendizado de música tendem a se beneficiar dessa nova tecnologia tão breve a mesma encontra-se disponível para isso. Este trabalho apresentou as questões envolvidas na criação de ecossistemas de Internet das coisas musicais. Estas questões envolvem diversos desafios, que passam tanto pela necessidade de desenvolvimento tecnológico quanto pela necessidade de repensar questões

artísticas de maneira a permitir a criação artística que utilize esta tecnologia. Também é possível visualizar que, como todo avanço tecnológico, o surgimento desta tecnologia poderá trazer também questões sociais e econômicas que podem se tornar desafios sobre os quais é necessário nos debruçarmos em nossas pesquisas futuras.

8. Agradecimentos

Agradecemos ao CNPq e à CAPES(88887.486097/2020-00), PROPE/PROEX/PROA da UFSJ por financiar esta pesquisa e também a todos os membros do ALICE que contribuíram para a evolução desse trabalho.

Referências

- [1] Talita SINGER. Tudo conectado: conceitos e representações da internet das coisas. *Simpósio em tecnologias digitais e sociabilidade*, 2:1–15, 2012.
- [2] Alexandre Silva de Almeida. *Protocolos e Aplicações para Internet das Coisas Musicais*. Dissertação (mestrado em ciência da computação), Universidade Federal de São João del-Rei, 2020.
- [3] Luca Turchet, Carlo Fischione, Georg Essl, Damián Keller, and Mathieu Barthet. Internet of musical things: Vision and challenges. *IEEE Access*, 6:61994–62017, 09 2018.
- [4] Liam Lacey. Como a internet das coisas poderia impactar composição musical, produção, 2015.
- [5] Luca Turchet, Carlo Fischione, and Mathieu Barthet. Towards the internet of musical things. *smc2017*, pages 1–8, 2017.
- [6] Plínio Martins Filho. Direitos autorais na internet. *Ciência da Informação*, 27(2), 1998.
- [7] Ricky O'Bannon. Why virtual reality and a symphony orchestra are a natural fit, 2015.
- [8] Ricky O'Bannon. Virtual reality at the orchestra, 2018.
- [9] Cristina Rottondi, Chris Chafe, Claudio Allocchio, and Augusto Sarti. An overview on networked music performance technologies. *IEEE Access*, 4:8823–8843, 2016.
- [10] John Lazzaro and John Wawrzynnek. A case for network musical performance. In *Proceedings of the 11th international workshop on Network and operating systems support for digital audio and video*, pages 157–166. ACM, 2001.
- [11] George Xylomenos, Christos Tsilopoulos, Yannis Thomas, and George C Polyzos. Reduced switching delay for networked music performance. In *Packet Video Workshop (Poster Session)*, 2013.
- [12] Flávio Luiz Schiavoni, Marcelo Queiroz, and Marcelo Wanderley. Network music with medusa: A comparison of tempo alignment in existing midi apis. In *Proceedings of the Sound and Music Computing Conference*, Stockholm, Sweden, 2013.
- [13] Rômulo Vieira, Luan Luiz Gonçalves, and Flávio Luiz Schiavoni. The things of the internet of musical things: defining the difficulties to standardize the behavior of these devices. In *2020 X Brazilian Symposium on Computing Systems Engineering (SBESC)*, pages 1–7, 2020.
- [14] Rômulo Vieira, Mathieu Barthet, and Flávio Luiz Schiavoni. Everyday use of the internet of musical things: Intersections with ubiquitous music. In *Proceedings of the Workshop on Ubiquitous Music 2020*, pages 60–71, Porto Seguro, BA, Brasil, November 2020. Zenodo.

CongAr: uma proposta de instrumento musical digital baseado em congas percussivas

Lucas Zampar¹, Josemar Gama¹, Joelden Lopes¹, Douglas Silva¹, Felipe Rodrigues¹, Cláudio Gomes¹

¹Departamento de Ciências Exatas e Tecnologias - Universidade Federal do Amapá (UNIFAP)
Caixa Postal 261 – 68.906-970 – Macapá – AP – Brazil

{lucas.26.zampar, josemargama7, joelden.ap, douglasdasilvasousa8, leepe.fr}@gmail.com

claudiorogério@unifap.br

Resumo. Em recentes anos, é possível observar a proliferação de novos instrumentos musicais digitais (IMD) com diferentes desempenhos e finalidades. Atualmente, os IMD possuem combinação de interfaces de síntese sonora, gestuais e demais sensores com possível incorporação aos computadores. Este trabalho propõe um protótipo de instrumento musical digital para percussão em congas intitulado de CongAr. O CongAr utiliza dois sensores de distância representando as congas e um protótipo de arduino, ou seja, a cada distância captada pelo sensor transforma-se em um golpe musical de tambor. O CongAr transmite informações por modo em série, rede de comunicação local ou global em forma sonora ou MIDI. O CongAr é um modelo distribuído performático, que permite ações entre vários participantes podendo tocar em conjunto. Para isso, o CongAr possui um gerenciador de captação de informações para gerar os referidos conteúdos. Realizou-se testes com Músicos e não-músicos a partir de um questionário qualitativo apresentando satisfatório resultados de usabilidade, entendimento e comunicação visual.

Palavras-chave — instrumento musical digital, movimento e gestos, sistemas interativos em tempo real

Abstract. In recent years, it is possible to observe the proliferation of new digital musical instruments (DMI) with different performances and purposes. Currently, DMIs have a combination of sound synthesis interfaces, gestures and other sensors with possible incorporation into computers. This work proposes a prototype of a digital musical instrument for congas percussion entitled CongAr. CongAr uses two distance sensors representing the congas and an arduino prototype, that is, each distance captured by the sensor turns into a musical drum stroke. The CongAr transmits information via serial mode, local or global communication network in audio or MIDI form. The CongAr is a distributed performative model, which allows actions between multiple participants and can play together. For this, the CongAr has an information capture manager to generate the referred contents. Tests were carried out with musicians and non-musicians from a qualitative questionnaire with satisfactory results of usability, understanding and visual communication.

Keywords — digital musical instruments, movement and gesture, Real-time Interactive Systems

1 Introdução

Em recentes anos, é possível observar a proliferação de novos instrumentos musicais digitais (IMD) com diferentes desempenhos

e finalidades. Com esse avanço, há várias soluções que auxiliam o músico em suas composições ou ações artísticas [1].

Dependendo do discurso musical, o músico pode utilizar de várias ferramentas: instrumentos (violão, saxofone, trompete, flauta, etc.); instrumentos digitais (teclados sintetizadores, percussão, bateria digital etc.). Dessa forma, o processo de criativo continua em evoluções permitindo maior clareza e afinidade do músico à tecnologia.

Inseridos no cotidiano da sociedade atual, há ainda a ubiquidade musical que a partir de pequenos dispositivos como os microcontroladores e sensores, possuem poder computacional para síntese sonora, mobilidade, conectividade com outros dispositivos e com custo acessível. Além disso, as inúmeras ferramentas computacionais de processamento em tempo real existentes facilitou o desenvolvimento de aplicações personalizadas e não-comerciais com o de uso o desenvolvimento de protótipos com linguagem aberta como Arduino, Raspiberry Pi, Orange Pi etc. [2, 3]. Assim, permitiu-se a possibilidade de práticas de composições e desenvolvimento de instrumentos por meio de tecnologias, intensificando ainda mais a busca por soluções, modelos e ações para apresentações de artísticas ou musicais.

Este trabalho apresenta um protótipo de instrumento musical digital para percussão em congas intitulado de CongAr. O CongAr utiliza dois sensores de distância representando as congas e um protótipo de arduino. Cada sensor de distância é responsável por emular a funcionalidade de um tambor de conga. Assim, a cada distância captada pelo sensor transforma-se em um golpe musical que pode gerar informações sonora e MIDI. A comunicação do CongAr pode ser em modo serial, conectividade sem fio em rede local ou através de internet.

O artigo está organizado na seguinte forma. A seção 2 apresenta discussões de trabalhos relacionados. Na seção 3, apresenta-se o protótipo CongAr. A seção 4 apresenta os resultados obtidos e por fim a seção 5, comenta-se as conclusões e trabalhos futuros.

2 Trabalhos Relacionados

Ferreira e Schiavoni (2018) [4] implementaram um instrumento digital cujas interfaces de entrada são um controle de videogame dual analógico ou uma câmera de vídeo. Apesar do emprego de tais dispositivos com essa finalidade não ser inédita, eles são encontrados facilmente a um baixo custo. Ainda que disponha de muitos botões, o controle guarda a desvantagem de não possuir sensores de pressão neles que poderiam relacionar com o atributo de velocidade ou intensidade das notas. Já a câmera pode levar a um alto processamento, dado o emprego de algoritmos de visão computacional, o que pode comprometer a experiência de fluidez. Como sintetizador, empregaram uma solução já existente compatível com o protocolo MIDI, porém tal escolha pode restringir a liberdade na exploração de parâmetros de síntese.

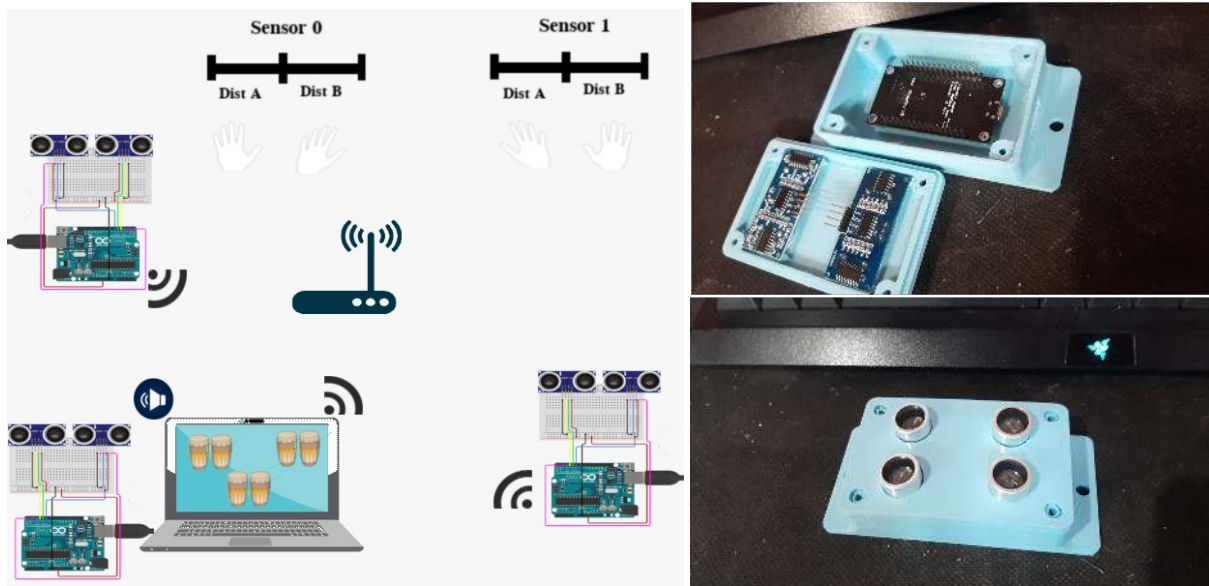


Figura 1: Visão geral do protótipo CongAr

Em relação ao mapeamento, focaram no processo criativo que despertasse o interesse e satisfação do usuário ao tocar. Nesse contexto, realizaram três mapeamentos para o controle permitindo diferentes níveis de interação musical com seus respectivos níveis de dificuldade. Já para a câmera, optaram por dois mapeamentos. Diante disso, conclui-se que é possível explorar dispositivos ubíquos, como controles e câmeras, com o intuito de servirem como interfaces de entrada para o desenvolvimento de IMD, permitindo diferentes interações do usuário por meio da escolha do mapeamento adequado.

O *TorqueTuner* é um módulo de mapeamento mecânico para IMD [5]. *TorqueTuner* tem o objetivo de simular a sensibilidade tátil associado com botões rotativos. Permite o mapeamento de sensores com parâmetros táteis e ajustes dinâmicos conforme a força de ação empregada pelo participante em tempo real. O controle angular de ações dos participantes é estimado a partir de sensor magnético que inclui a derivada primeira e segunda ordem para calcular velocidade e aceleração. Utilizaram um sistema com o microcontrolador ESP32, e célula de bateria *LiPo*. Realizaram implementações ao *T-Stick* em programação no ambiente programacional do *SuperCollider*. Apesar do *TorqueTuner* apresenta-se como um módulo robusto. Abre grandes discussões sobre variações de criatividade artísticas e ainda não permite a interatividade entre participantes.

O trabalho de Turchet e Barthet desenvolveram uma guitarra eletrônica sensoriada para interatividade com tecnologia de conectividade na nuvem e sem fio local [6]. Desse modo, a guitarra pode ser interagida com os aplicativos dedicados com diversas funções de interações. Apesar de suas ações apresentarem características inclusivas, têm efeitos únicos e diferentes, ou seja, é necessário definir os instrumentos que faram parte das ações em conjunto.

3 CongAr

O CongAr é um protótipo de um instrumento musical digital (IMD) baseado em tecnologia da Internet das Coisas Musicais (IoMusT) [7]. As subseções a seguir apresentam as funcionalidades e características do protótipo CongAr. A subseção 3.1, descreve as funcionalidades do protótipo e a subseção 3.2 apresenta características do ambiente visual.

3.1 Funcionalidades de comunicação e informação

O CongAr funciona no modelo de comunicação cliente-servidor, conforme a figura 1. O cliente é composto por dois sensores de distância US-100 conectados em um arduino no modelo ESP8266, ou seja, cada instrumento de congas é baseado em um arduino. Assim, o arduino capta os dados de distâncias dos sensores e transmite para o servidor.

O servidor é responsável em mapear os dados recebidos de cada cliente transformando-as em informações sonoras ou MIDI. O servidor emula um instrumento para cada arduino em comunicação com o servidor. O servidor oferece opções para o gerenciamento de timbres de congas para cada cliente, portanto, cada arduino possui identificação única e automática.

O CongAr cliente utiliza dois sensores US-100 em cada arduino simulando a conga aguda e grave. Os sensores são independentes, ou seja, o sensoriamento das distâncias não possuem correlação ou ainda dependência para a geração de diferentes opções musicais. Dessa forma, mapeou-se as tonalidades abertas e fechadas para cada sensor, ou seja, o *sensor 0* pode gerar informações de conga aguda com tonalidade aberta ou fechada e o *sensor 1*, conga grave com tonalidade aberta e fechada. O sensor captando dados de distância dentro da área A, indica a tonalidade fechada e, dentro da área B, a tonalidade aberta. Assim, pode-se gerar as informações de conga: aguda-fechada (sensor 0 - dist A), aguda-aberta (sensor 0 - dist B), grave-fechada (sensor 1 - dist A) e grave-aberta (sensor 1 - dist B).

O servidor e o cliente CongAr podem estar conectados via comunicação serial, em uma mesma rede local sem fio, ou ainda na utilização da opção de conexão por Internet, o servidor obviamente deverá ter conexão com a Internet, assim como o cliente. Para a opção de conexão por Internet, tanto o servidor quanto o cliente não necessitam de restrições de localidade. Para a opção de conexão via Internet, utilizou-se o protocolo *MQTT* analisando-se os critérios de QoS de jitter e delay obtendo bons resultados para prioridades de comunicação alta e média [8].

3.2 Ambiente visual

Além de captar dados de distâncias de cada arduino e transmitir informações sonoras e MIDI, o servidor gerencia o ambiente vi-

sual, conforme a figura 1 para três participantes simultaneamente. O ambiente visual é responsável em representar movimentações gestuais de conga transformando-as em informações sonoras e MIDI. Os gestos podem ser captados por qualquer parte do corpo ou objetos que funcionem como barreira sensorial para o modelo US-100. Assim, é possível visualizar os golpes musicais por cada participante.

O ambiente visual foi desenvolvido pela biblioteca Pygame [3] para a geração de informação sonora e MIDI. Desse modo, para cada participante é emulado um novo instrumento que transmite informações MIDI em tempo real para qualquer programa de produção musical como LMMS [9], Ardour [10] etc.

Para a informação sonora, utilizou-se de uma base de dados gratuita com os sons de conga-grave e conga-aguda, com os golpes musicais aberto e fechado [11]. Em relação à informação digital MIDI, utilizou-se as numerações 69, 70, 71 e 72 para a identificação dos golpes musicais de *grave-aberto*, *grave-fechado*, *agudo-aberto* e *agudo-fechado*, respectivamente.

3.3 Usabilidade

O participante pode interagir com o CongAr cliente dispondo-o em uma mesa, no solo ou em ambiente livre de objetos que interfiram na captação de distâncias. Devido às condições dos sensores, a distância ideal de cada participante na usabilidade do CongAr cliente é de aproximadamente 50 cm. Conforme a figura 1, mapeou-se a área A, de 10 cm até 40 cm para captar ações de congas graves. A área B ficou mapeada para a partir de 45 cm em até 80 cm.

Percebeu-se distorções de dados de distâncias captados acima de 130 cm. Dessa forma, pode-se adicionar variações de golpes musicais sem ter perda de qualidade de captação de dados de distância. Assim como, pode-se, futuramente, modificar as sonoridades abertas e fechadas para demais variações necessárias para cada participante.

4 Resultados

Realizou-se a avaliação sobre o quesito de criatividade do protótipo CongAr a partir de entrevistas com participantes divididos entre 10 músicos e 10 não-músicos. Considerou-se músicos, participantes com algum conhecimento ou contato com teoria ou instrumentos musical. Os participantes improvisaram o uso do CongAr por dois minutos sem alguma informação inicial e responderam um questionário de 4 perguntas na escala Likert de 1 a 5, seguindo a ordem crescente de concordância. Originalidade do resultado, fácil colaboração, atenção do participante e fácil usabilidade, foram os fatores utilizados no questionário. Vale ressaltar que as perguntas são baseadas no modelo de caracterização de instrumentos digitais utilizadas [12].

A figura 2 apresenta os resultados em relação ao fator de criatividade com relação aos entrevistados comparados entre músicos e não-músicos para cada fator, considerando as informações de média e desvio padrão. De maneira geral, os resultados para não-músicos ficaram com maior índice de aceitação comparado com os músicos. Este resultado inferi-se que músicos necessitam explorar todas as possibilidades do protótipo CongAr, o que não aconteceu com não-músicos.

A maior e menor diferença de respostas médias entre músicos e não-músicos foi a usabilidade e atenção do participante, respectivamente. Analisando apenas os entrevistados considerados não-músicos, o fator de avaliação atenção do partici-

pante e usabilidade obtiveram a menor e maior média com valores de 4,19 e 4,89, respectivamente. Com a pequena variação dos resultados, percebe-se que os não-músicos conseguiram gerenciar o CongAr e assim, desenvolver de alguma forma intuitiva habilidades musicais.

Para os músicos entrevistados, o fator de fácil manuseio apresentou a maior média com valor de 3,93 e, o fator de usabilidade, a menor média em 2,81. Dessa forma, percebe-se que músicos conseguiram bom manuseio do protótipo, realizando variações de novas possibilidades criativas, porém não tiveram bom êxito em novas opções ou descobertas independentes. Ao final, os músicos tiveram críticas construtivas com boas sugestões de aprimoramento do protótipo. Cada sugestão é totalmente aceitável, porém estaria diretamente relacionada para um determinado evento, dessa forma as sugestões ficam a necessidades ou o perfil de cada participante em adicioná-las.

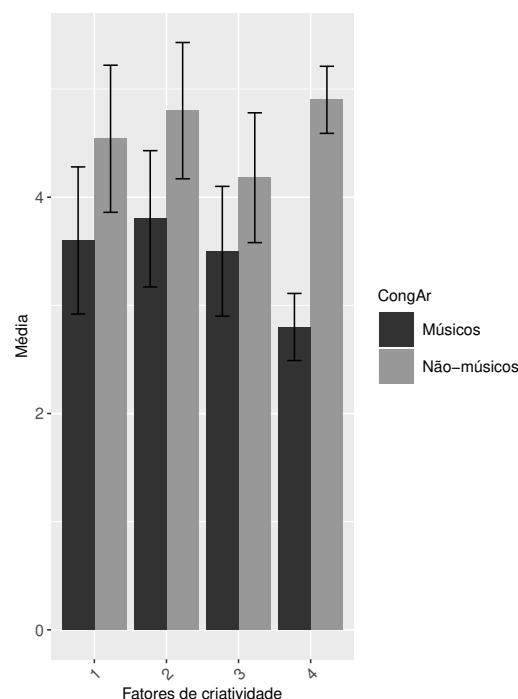


Figura 2: Resultado da análise de criatividade

5 Conclusões e trabalhos futuros

Há inúmeras possibilidades após a construção deste protótipo, por exemplo, no auxílio em escolas musicais com o instrumento de percussão; geradores de frases musicais para congas; banda de performance de congas digitais; e ainda, ultrapassar as barreiras locais enviando os dados do protocolo MIDI via rede 802.11 ou ethernet. Conclui-se que, o protótipo CongAr pode ser uma ferramenta com diversas possibilidades de utilização aplicadas em escolas, estúdios e performance musicais, todas pré-ajustadas conformes suas necessidades e objetivos de cada participante.

Referências

- [1] Koray Tahiroğlu, Thor Magnusson, Adam Parkinson, Iris Garrelfs, and Atau Tanaka. Digital musical instruments as probes: How computation changes the mode-of-being of musical instruments. *Organised Sound*, 25(1):64–74, 2020.
- [2] Simon Monk. *Movimento, luz e som com Arduino e Raspberry Pi*. Novatec Editora, 2019.

- [3] Pete Shinnars. Pygame. *Dostupné z:* <http://pygame.org/>[Online, 2011].
- [4] Gabriel Lopes Rocha, Avner Maximiliano, João Pedro Moreira Ferreira, and Flávio Luiz Schiavoni. Desenvolvimento de instrumentos musicais digitais a partir de dispositivos ubíquos.
- [5] Mathias Kirkegaard, Mathias Bredholt, Christian Frisson, and Marcelo Wanderley. Torquetuner: A self contained module for designing rotary haptic force feedback for digital musical instruments. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 273–278, 2020.
- [6] Luca Turchet and Mathieu Barthet. An ubiquitous smart guitar system for collaborative musical practice. *Journal of New Music Research*, 48(4):352–365, 2019.
- [7] Luca Turchet, Francesco Antoniazzi, Fabio Viola, Fausto Giunchiglia, and György Fazekas. The internet of musical things ontology. *Journal of Web Semantics*, 60:100548, 2020.
- [8] Roger A Light. Mosquitto: server and client implementation of the mqtt protocol. *Journal of Open Source Software*, 2(13):265, 2017.
- [9] Linux Multimedia Studio DigitalOcean. Disponível em: <https://lms.io/>. Acesso em 20.07.2021.
- [10] Ardour Paul Davis. Disponível em: <https://ardour.org/>. Acesso em 20.07.2021.
- [11] et al. Zynaddsubfx musical synthesizer N. O. Paul, M. McCurry. Disponível em: <http://zynaddsubfx.sf.net/>. Acesso em 20.07.2021.
- [12] Atau Tanaka, Alessandro Altavilla, and Neal Spowage. Gestural musical affordances. *Sound and Music Computing*, 2012.

Lab Reports

A prospective report on the research developed at the Laboratory of Audio and Music Technology at USP

Regis Rossi A. Faria, Ricardo Thomasi, João Monnazzi, Eduardo Bonachela, André Giolito, Gabriel Lemos

Laboratory of Audio and Music Technology (LATM) – School of Arts, Sciences and Humanities of the
University of São Paulo (USP)
Rua Arlindo Bettio, 1000 – 03828-000 São Paulo, SP

regis@usp.br, ricardothomasi@usp.br, joao.monnazzi@usp.br, eduardobonachela@usp.br,
andregiolito@usp.br, gabriel.lemos@usp.br

Abstract

This paper presents a concise report on the research developed at the Laboratory of Audio and Music Technology (LATM) at the School of Arts, Sciences and Humanities of the University of São Paulo (EACH-USP). The laboratory was founded in 2011 targeting the areas of music technology, musical acoustics and bioacoustics, strengthening its scope in 2019 to the areas of sound and music computing and audio engineering. Six projects are presented herein, describing their application areas, goals, achievements and perspectives.

1. The laboratory history and research orientation

The Laboratory of Audio and Music Technology (LATM) started its trajectory in 2011 in the Music Department at USP Ribeirão Preto campus. Counting on an infrastructure combining a technical room adjacent to a performance and recording room in a building designed for the department, the initial focus was to support under-graduation education and to promote research lines in musical acoustics and music technology at the campus.



Figure 1: A view of an event at LATM studio

A research emphasis on auralization, surround sound and musical spatialization was present from the beginning, counting with available sets of loudspeaker arrays and multichannel interfaces. In eight years the laboratory successfully implemented a series of educational activities, such as the LATM seminars on musical acoustics and audio technologies, and had several

scholarship students in scientific initiation programs. Along the years it held several scientific events, including an AES (Audio Engineering Society) regional meeting on new frontiers in musical instrumentation (2014), and introduced innovative research lines, resonating with both internationally relevant and active research topics and local scientific influences, for instance on chemistry and bioacoustics.

Supporting the investigation lines and activities of its main researchers, the laboratory became home of the AUDIENCE spatial audio project and started the Reactive Music project, an innovative long-term research exploiting new sound processing concepts. Starting in 2016, a prospective collaboration with the Laboratory of Ethology and Bioacoustics (EBAC) at FFCLRP-USP opened new frontiers for research in the analysis of the vocalization of terrestrial mammals and in the design of monitoring devices and proposals for bioacoustic inference.

In 2019 the laboratory underwent a shift to another campus in São Paulo, strengthening the research orientation to sound and music computing and audio technology. The laboratory is currently working on the design of a new infrastructure to accommodate its activities, while articulates with local communities demands and collaborates with leading research laboratories at USP, such as LabArteMídia (ECA-USP), CITI-USP (EP-USP), EBAC (FFCLRP-USP), NuSom (ECA-USP), and at other universities, such as UNICAMP.

2. Lab projects

This section presents the main projects currently under development at the laboratory. Some of them are active since the foundation of the laboratory, characterizing long term research lines. Most projects have been investigating specific fronts along the years, connected to graduation and post-graduation research goals.

2.1 Acoustic-digital feedback control strategies for electroacoustic performance

This research investigates music structural possibilities in emergence theories aiming at electroacoustic performance. A fertile investigation

territory has been uncovered in paradigms that includes the acoustic environment as part of the musical system. In this sense, the *Audible Ecosystem* paradigm [1] and the concept of *Performance Ecosystem* [2] were taken as analytical scope, because they drive a peculiar perspective of acoustic and digital feedback instrumentalization as a principle to sound emergent structures generation and interface of interaction as well.

Four musical studies under development, called *Ecos 1-4*, had contributed as an experimentation territory for audible ecosystems observation, for testing instrumentalization feedback techniques and musical structuring. An acoustic-digital feedback control Max/Msp patch was developed based on an adaptive multiband gain control engine (AMGC), enabling an intuitive and responsive interaction with the resonant frequencies, favoring improvisation and multiple performance situations, such as laptop music, acoustic and electric instruments, body performances and interactive sound installations (Figure 2).

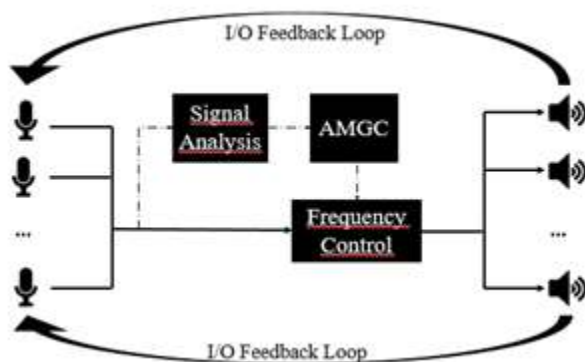


Figure 2: Digital-acoustic feedback loop control [9].

In this project we have been following uncovered strategies so far, such as (a) a new approach to multichannel sound spatialization, which in turns starts to have an impact on the emergent spectra morphology [3]; (b) other perspectives of digital filtering for sound synthesis; (c) strategies for instrument incorporation as ecosystem's part, under development in *Study for Ecosystemic Guitar* [4]; (d) a theoretical-analytical framework that relates the audible ecosystem's behavior and the sound patterns with a feedback and causal correlation-based taxonomy of emergencies.

From this, one hopes to draw a novel music structural thinking that may be helpful in situations where sounds are emergencies from performance interactions, and they are not known in advance [5].

2.2 Reactive Music - sound processors for reactive musical applications

A sound processor that uses stoichiometry chemical concepts metaphorically in order to produce new sounds out of the reaction between two reagent sounds is the core rationale in this project. This research is within the realm of the attractive Intelligent Musical Production research branch, towards interesting techniques and automations for sound mixing, mastering

and effects.

To explore and to apply the chemistry knowledge to process real-time musical performance we have been designing, implementing and testing a suite of tools in the form of Pure Data (Pd) [6] patches, abstractions and externals.

The current package contains the following main sound processors: *timepitch* (an independent time and frequency transformer), *funew* (a band-pass filter for harmonic spectra) and *reator* (an environment to host musical reactions). Along the project phases, a number of auxiliary functions required to build the main sound processors were not available elsewhere, and were then developed, such as: *antideadlock* (a tool for consistent linkage between numbers and sliders); *faixa* (a numerical range converter), *instr* (a simple additive synth for testing processor behaviors) and *patchbay* (an output converter to several multichannel/surround formats).

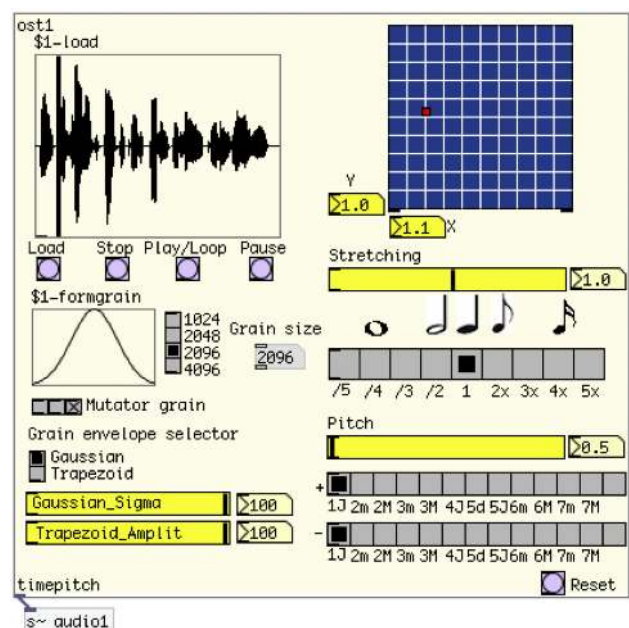


Figure 3: TimePitch sound processor

Figure 3 shows, for instance, the *timepitch* interface. This processor is built upon granular synthesis and can perform independent time stretching and pitch transposition with time steps in standard music note durations and frequency steps in standard music intervals.

Taking into account the progress achieved on interfaces and sound processing schemes [7], we have proposed a *musical reactor equation* that defines: (1) the quantities of each sound reagent that will react to each other, and (2) the functional reaction type.

The first type of processing explored was the mixing functional. The reactive mixing can be thought of as a mixing method in which the flow of reagent sounds is controlled according to a reaction equation, weighted by reagent parameters. We expect to use this material to advance in developing other types of reaction using ring modulation, convolution and morphing functionals. We are also aiming at building computationally effective plug-ins, since most sound processing tools require

demanding computer resources.

A set of control interfaces, spectral/time processors, auxiliary objects and versions of the reactor were developed already. A prototype of the first plug-in was recently brought up using a compiled version of the Camomile VST3 effect [8] along with specific externals used in the reactor encoded in C++. It behaves as a host so as to enable the patch to become a plug-in itself.

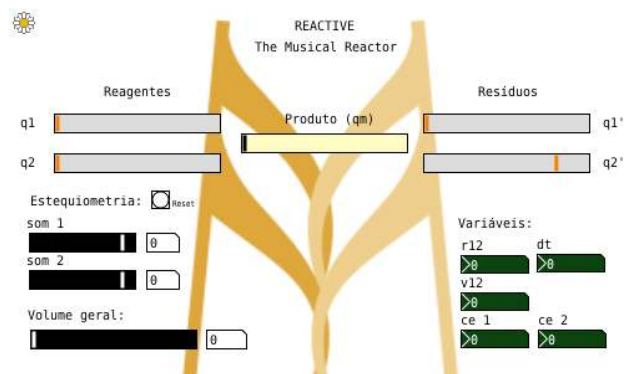


Figure 4: Reactive-camomile VST3 prototype interface.

The group is currently setting up a Github repository to provide the library distribution with free availability to benefit a community of musical artists, sound producers and anyone keen on musical technology. A next step in the project shall be a Pd-independent plug-in version, using for instance the JUCE framework and C++ encoding.

2.3 Musical Stereotype: analysis of museme recurrency in film scoring

A reiteration of musical elements or structures that we call musemes [9] is identified in cinematographic musical tracks, which, linked to certain archetypes or moods, and reiterated over decades in similar connotative senses, gained extra-musical meanings. The characteristics of these recurrences in sound cinema, however, are not well known. We propose that these musemes are identifiable and extractable, and that this search can be optimized with the help of computational tools, such as musical and symbolic descriptors.

This project proposes an investigation on musemes reiterated in specific cinematographic genres, raising the quantity and the way in which these musical materials were used and the way in which they were linked to the images. By producing a mapping of examples and analysis of the recurrence of these musical stereotypes, it is intended to provide support material for audiovisual professionals and researchers for the study and creation of sound narrative in Western cinema.

It is our hypothesis that Photoplay Collections recurrently make use of themes, topics and musical structures associated with moods, archetypes and scenes, which we call musemes, and can thus be considered the first musical materials created specifically to accompany cinematographic presentations and with a defined

intention of contribute to the filmic narrative. Following this perspective, we assume that the musemes in the cinematographic film scoring originated in musical collections, with the focus of the project being an investigation of the occurrence and reiteration of these structures in a defined cut of original musical tracks, created by several renowned composers of Hollywood cinema.

We propose an unprecedented tracking of cinema musemes using the method of interobjective comparison [10] aided by symbolic and musical descriptors, where we relate musical structures, their similarity and uses in the scene. We aim to track, qualify and quantify the recurrence of certain musemes, associated with specific archetypes and moods, within sound movie years (1930 to 2020), from a selection of films proposed for analysis.

Our specific objectives are defined strategies for museme description; toolkit development and validation for museme identification and characterization through music and symbolic descriptors, evaluate the currency of musemes appearance on film scorings.

Figure 5: Interobjective comparison between the themes Mysterious (1914) (top) and Imperial March (1970) (bottom).

So far, we analysed four photoplays collections and tracked six musemes that can be linked to the shadow

archetype [11]. Through a manual analytic method we found that musemes in the *Imperial March*, music theme of Darth Vader from Star Wars. This preliminary result shows that our hypotheses make sense and now we are gonna enlarge our number of analyses by using the symbolic descriptors Music21 [12] and JSymbolic [13]. By using these tools to improve and accelerate our analytic method, we expect to cover a number of film scores that allows us to have data that can answer us if there is a reiteration of the musemes found in the photoplays and, if so, what is the frequency in which they appear on movies.

2.4 Bioacoustic inference - sound analysis for animal identification and respiratory-based audio diagnosis

In the context of COVID-19 pandemic, fast and at-home identification of symptoms that require medical care by non-invasive diagnosis and without social exposure can be a very useful tool to ensure quick action in potentially severe cases of respiratory problems.

In this project, we seek to research descriptors and schemes for inference in bioacoustics to analyze recordings of respiration of healthy individuals infected by the new coronavirus made by cell phones at different stages of infection, and to verify the emergence of respiratory patterns indicative of attention. One step is to validate the effectiveness of analysis and classification techniques for differential audio-diagnostics and then, at a later stage, propose a tool such as a website or application for widespread use in society.

Original recordings exhibit different qualities, noise floor, velocities and number of cycles, and it is required a normalization procedure prior to proceed to content analysis. We have been working on such a pre-processing for a number of respiratory recordings.

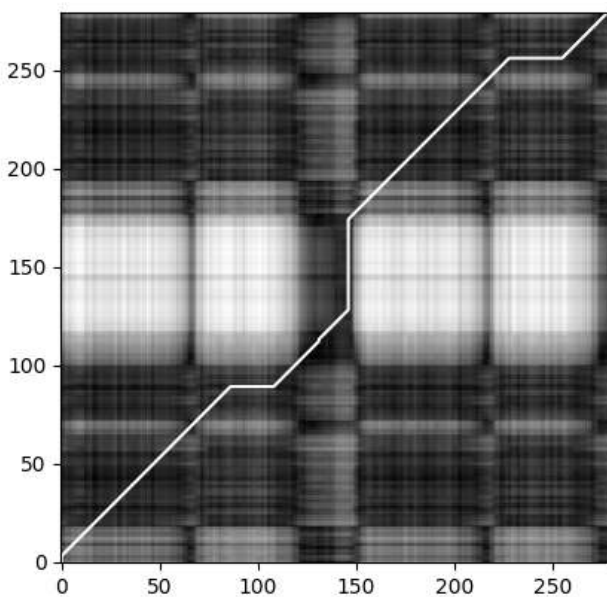


Figure 6: Comparison using MFCC + DTW between a normal respiratory sound (vertical axis) and one with crackle noise (horizontal axis)

After this step, we have been investigating the use of Mel-frequency cepstrum coefficients (MFCC) [14] and Dynamic time warping (DTW) [15] to analyze the sounds and evaluate its efficacy for comparison and classification. A library of reference sounds include normal respirations, and disease examples, such as Littmann Library [16] and ThinkLabs [17]. Figure 6 shows the maximum cost matrix between two respiratory sounds, and shows a white line indicating the shortest path between the two sequences, which can be an element that would allow classifying the data.

2.5 AUDIENCE - technologies for immersive audio (2D/3D sound)

AUDIENCE and OpenAUDIENCE are versions of a spatial audio software library being developed since 2005. The initial project and the current distribution of the programs is hosted at www.lsi.usp.br/audience. Since then, its unique functional architecture has been inspiring subsequent research projects, gathering a number of collaborators in what can be seen as a systematic effort to investigate and develop technologies for the production and reproduction of 2D/3D audio and immersive sound for applications in music, acoustics, audio engineering and bioacoustics [18] [19] [20].

Its development has been done in "periodic waves", with each phase addressing one spatial audio technology or a problem to solve. First wave addressed the Ambisonics in CAVES and open-spaces, issues on source localization and directivity, then exploring the codecs integration such as MPEG SAOC and MPEG Surround, then sound scene auralization for musical and architectural scenes, multichannel and microphone array-based recordings, then wave-field synthesis and lately addressing 360/3D sound scene production and rendering using HOA and object-based strategies.

The system operation is based on concentrating all operations of the spatial sound production and reproduction chain in 4 functional layers, and all processes and functions follow this design architecture.

Resulting hardware products for spatial audio include: our own-developed Wave Field Synthesis (WFS) auralization rig with 18 linear loudspeakers, hosted at the partner-lab NEAC (Audio Engineering and Coding Center at EP-USP); a bi-hemispheric 360° sound scene capture board to which 2 portable recorders are attached; and a 3D microphone prototype shown at Figure 7.



Figure 7: A 3D microphone array prototype with 6 orthogonal capsule

Software developed for spatial audio include the OpenAUDIENCE library: a whole distribution of patches and objects for sound immersion, auralization, and surround sound in Pure Data (Pd) [21]; solutions for the production and reproduction of spatial sound scenes (SSS); a WFS auralization engine; Figure 8 shows the main interface of the Spatial Sound Scene producer module.

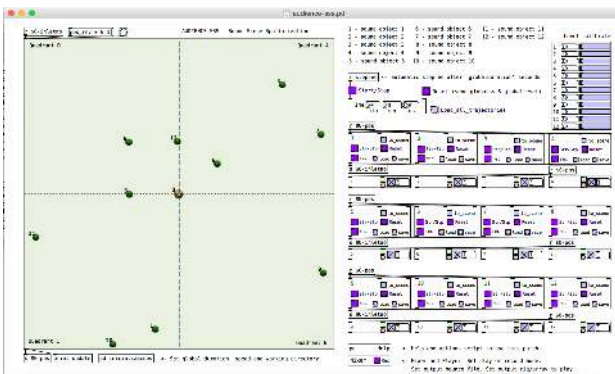


Figure 8: The Spatial Sound Scene renderer module

The community of spatial audio research is used to change its approach target from time to time, sometimes revisiting prior topics under a different convention, adding more knowledge and new products to them. Recently developments on higher order Ambisonics (HOA) and object-oriented audio scene rendering have been incorporated in 360 degree audiovisual media and adopted by the market, as in Facebook and YouTube.

Together with LabArteMidia (ECA-USP) we have been exploring the making-of for such spatial narratives and developing techniques for sound scene production and for rendering such immersive media formats. This scenario has been strongly influenced by extended reality (XR) and pulled by MPEG standards, such as the MPEG-H (3D Audio) and the forthcoming MPEG-I, targeting immersive and augmented reality. This has shown to be a trend for the new generation of digital TV worldwide, including in Brazil [22].

2.6 The spectrum in the machine: use of neural networks for sound synthesis in contemporary compositional contexts

Considering the growing use of machine learning in digital artistic practices in the last decade, especially its improvements in complex sonic style transfer techniques and sound synthesis, this research proposes a comparative study between different topological structures of Neural Networks applied to music generation processes in the waveform domain.

We propose an evaluation of the state of the art of these technologies in the field of contemporary sound creation, and we aim to identify the technical limitations and aesthetic possibilities for the application of neural networks in multimedia composition contexts. The method considers the elaboration of authorial artistic processes guided by successive generative experiments - especially in the implementation of Recursive Neural Networks (RNN) and the Generative Adversarial Networks (GAN) models [23].

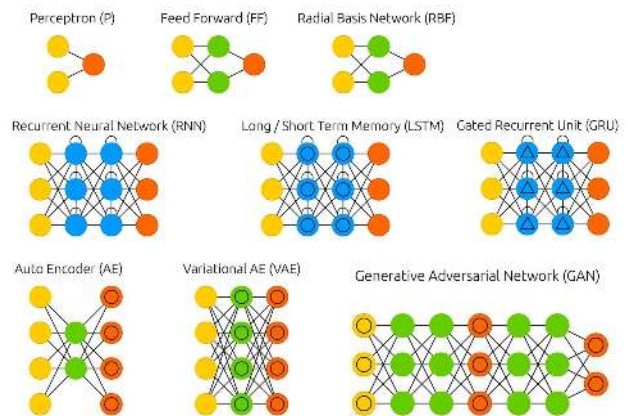


Figure 9: Graphic representation of Perceptrons, RNN and GAN topological structures

An expected result is to document in detail the creative process and the functioning of this technology so that, at the end of the research, we can make available programs dedicated to music synthesis using open-source software and code libraries that could then be accessed online and used by active developer communities.

The relevance of the research in the implementation of these models in the field of sound creation for the Brazilian context focuses on the study of the adequacy of machine learning techniques in the music synthesis process and its broader political and economic implications for the use of this technology.

The project is currently in its initial state, mainly focused in bibliographic revision for case studies and study of different end-to-end architectures based on RNN and GAN generative models implemented on Python. The implementation experiments will also be supported with additional computer resources by the Center for Artificial Intelligence (C4AI/Inova-USP) [24], in partnership with the activities held by the GAIA - Artificial Intelligence and Art Group (Inova-Usp) [25].

For the near future, we will be implementing

specific networks aiming for practical experimentations in compositional contexts.

3. Reflections and future trends

The diversity of themes and convergence in strategic areas of cultural, artistic, scientific, ecological and health interest has been a relevant characteristic of the group. The projects are aligned to the lab's main research topics legacy, which act as attractors, but are naturally connected to individual inspirations and interests brought by our researchers in post-graduation (masters and doctorate) or under-graduate (scientific initialization) programs. We shall acknowledge the importance of collaborating with other laboratories and centers, which is the actual key to real multidisciplinary research, inspiring applications, joining complementary competence and infrastructure.

Techniques for context-oriented sound analysis. The approach named 'bioacoustics inference' is a systematic pursuit of descriptive and classification metrics using multivariate analysis aiming to support ecologic demands about monitoring biomes and acquiring natural soundscapes information. Applications to support medical diagnosis for health. A new cycle has begun for explorations on music synthesis, beyond previous generations works on modular analog combinations and algorithmical deterministic digital synthesis. Frontiers for sound scene individualization, immersive audiovisual production, and 3D immersion with microphone and loudspeaker arrays, extended reality.

The current laboratory website is available at sites.usp.br/latm/ and it also has a Github repository at github.com/latm-lab/, which is to be one of the portals to disseminate the lab products. The knowledge sharing is also supported by the NEAC website at www.lsi.usp.br/neac.

References

[1] A. Di Scipio. Sound is the interface': from interactive to ecosystemic signal processing. *Organised Sound*, 8(3), 269–277. Cambridge University Press, 2003.

[2] S. Waters. Performance ecosystems: ecological approaches to musical interaction. *Electroacoustic Music Studies Network*. De Montford/Leicester: EMS, 2007.

[3] R. Thomasi and R. R. A. Faria. Moving along sound spectra: an experiment with feedback loop topologies and audible ecosystems. *International Computer Music Conference*. Santiago: ICMC, 2021.

[4] R. Thomasi and F. Kozu. Study for Ecosystemic Guitars: The electroacoustic improvisation in the sound emergence minefield. *The 21st Century Guitar: Unconventional Approaches to Performance, Composition and Research*. Lisboa: NOVA, 2021.

[5] R. Meric and M. Solomos, Agostino Di Scipio's music: emergent sound structures and audible ecosystems. *Journal of Interdisciplinary Music Studies*, 3 (1,2), 57-76, 2009.

[6] PUCKETTE, M. "Pd Documentation". http://crca.ucsd.edu/~msp/Pd_documentation/. Access: 23/07/2021.

[7] FARIA, R. R. A.; CUNHA JUNIOR, R. B. ; AFONSO, E. S. Reactive music: designing interfaces and sound processors for real-time music processing. In: *Proceedings of the 11th International Symposium on Computer Music Multidisciplinary Research (CMMR 2015)*, Plymouth, 2015. p. 626-633.

[8] P. Guillot. Camomile: Creating audio plugins with Pure Data. In: *Linux Audio Conference*, Berlin, Jun. 2018. Available at: <https://hal.archives-ouvertes.fr/hal-01816603>. Access: July 2021.

[9] P. Tagg and B. Clarida. *Music's Meaning: a modern musicology for non-musos*. Nova York: MMMSP, 2013.

[10] P. Tagg and B. Clarida. *Ten Little Title Tunes: Towards a musicology of the mass media*. The Mass Media Music Scholar's Press. Nova York and Montreal, 2003.

[11] C. Vogler. *The Writer's Journey: Mythic Structure for Writers*. New Frontier. Rio de Janeiro, 1998.

[12] M. Cuthbert and C. Ariza. Music21: A toolkit for Computer-Aided Musicology and Symbolic Music Data. 2010. *ISMIR, International Society for Music Information Retrieval*, 2010. pp. 637-642.

[13] C. McKay and J. Cumming. JSYMBOLIC 2.2: Extracting features from symbolic music for use in musicological and MIR research. 2018. *Proceedings of the 19th ISMIR COnference*, Paris, 2018. pp. 348-354.

[14] GORDILLO, Christian Dayan Arcos. Continuous speech recognition by combining MFCC and PNCC attributes with SS, WD, MAP and FRN methods of robustness. Thesis (Masters in Electrical Engineering), Pontificia Universidade Católica, Rio de Janeiro, 2013.

[15] W. Fu, X. Yang and Y. Wang, "Heart sound diagnosis based on DTW and MFCC," In: *2010 3rd International Congress on Image and Signal Processing*, 2010, pp. 2920-2923, doi: 10.1109/CISP.2010.5646678.

[16] Littmann Library. Lung Sounds, 2020, available at: <http://www.3m.com/healthcare/littmann/lung.html>. Accessed on July 5 2021.

[17] THINKLABS. Digital Stethoscope, 2020, available at: <https://www.thinklabs.com/>, accessed on July 5 2021.

[18] Faria, R.R.A. et al. AUDIENCE - Audio Immersion Experiences in the CAVERNA Digital. In: *Proceedings of the 10th Brazilian Symposium on Computer Music - Current Frameworks for Music Information Representation*, Belo Horizonte, 2005. p. 106-117.

[19] Faria, R.R.A et al, "Improving spatial perception through sound field simulation in VR," *IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems*, 2005, pp. 103-108, doi: 10.1109/VECIMS.2005.1567573.

[20] Faria, R. R. A. AUDIENCE for Pd, a scene-oriented library for spatial audio. In: *Proceedings of the 4th international Pure Data Convention*, Weimar e Berlin, 2011.

[21] OpenAUDIENCE library for sound immersion and auralization, v. 1.0.3 (2012), available in: <http://lsi.usp.br/neac/en/openaudience>

[22] Forum SBTVD, Call for Proposals: TV 3.0 Project, (17/07/2020), available at: <https://forumsbtvd.org.br/>

wp-content/uploads/2020/07/SBTVDTV-3-0-CfP.pdf

[23] VEEN, Fjodor Van. *The Neural Network Zoo*. The Asimov Institute, 2016. Published online at: <https://www.asimovinstitute.org/neural-network-zoo/>. Accessed on July 25 2021.

[24] Center for Artificial Intelligence, 2020, accessed on July 23 2021 at: <http://c4ai.inova.usp.br/pt/home-2/>.

[25] GAIA - Grupo de Arte e Inteligência Artificial, 2020, <https://sites.usp.br/gaia/>, accessed on July 23 2021.

Alice no país da pandemia (2021)

Flávio Luiz Schiavoni, Ana Clara Medina Fonseca, Avner Maximiliano de Paulo,
Carlos Eduardo Oliveira de Souza, Cleisson José Dias da Silva, Emerson Junio Silva Costa,
Fábio dos Passos Carvalho, Gabriel Lopes Rocha, Gabriel Rodrigues Chaves Carneiro,
Igino de Oliveira Silva Júnior, Igor Alves Estevam Andrade, Isadora Franco Oliveira,
João Marcos de Oliveira Lara, João Teixeira Araújo, João Pedro Mendes de Oliveira,
Jônatas Araújo da Silva, Luan Luiz Gonçalves, Lucas Estevão dos Santos,
Luiz Gustavo Colzani Monti Sousa, Mariana Pereira Lellis, Matheus de Bonfim Rodrigues Jordão,
Rafael Alves Soares de Andrade, Rebeca Lima Soares, Rômulo Augusto Vieira Costa, Samuel Rodrigues Rabay

¹ ALICE - Arts Lab in Interfaces, Computers, and Everything Else
UFSJ - Universidade Federal de São João del-Rei

<https://alice.dcomp.ufsj.edu.br/>

Resumo. O ALICE - Arts Lab in Interfaces, Computers, and Everything Else é um laboratório de pesquisa do Departamento de Computação da Universidade Federal de São João del-Rei. Acolhendo pesquisadores de várias áreas, o ALICE tem desenvolvido pesquisas nas áreas de processos criativos amparados pela tecnologia, música em rede, paisagens sonoras, orquestras de dispositivos e outras áreas em que é possível aliar arte e tecnologia sem que uma destas áreas se coloque acima da outra. Neste artigo, apresentamos nossos desenvolvimentos recentes e lamentos atuais em um ano em que a distância de nossos corpos tenta ser superada pela onipresença das redes de computadores.

1. Introdução

No departamento de computação da Universidade Federal de São João del-Rei encontra-se um laboratório que se dispõe a trabalhar as fronteiras entre arte e tecnologia. O Nome ALICE é acrônimo para Arts Lab in Interfaces, Computers and (Education, Exceptions, Experiences, Entertainment, Environment, Entropy, Errors, Everything, Else and Etcetera ...). Em português o laboratório poderia ser chamado de LAICO - Laboratório de Artes, Interfaces, Computação, e Outras coisas, o que não deixa de ser um bom nome. Este laboratório abriga alunos de cursos de graduação como Artes Aplicadas, Música e Computação e alunos de mestrado em computação e do programa interdisciplinar em Artes, Urbanidades e Sustentabilidade (PIPAUS).

A diversidade dos seus integrantes permite uma troca de experiência e saberes entre os alunos, além de permitir que o laboratório desenvolva pesquisas inter / transdisciplinares que envolvem Arte e Tecnologia sem que uma coisa se coloque acima da outra. Acreditamos que é possível pensar a transdisciplinaridade enquanto ponto de partida por envolver no nosso laboratório pesquisadores de diferentes áreas e com diferentes conhecimentos. Assim, em vez de tentar criar projetos inter/transdisciplinares, criamos projetos e convidamos pessoas de outras áreas para fazer parte destes projetos, sem ter certeza do que isso pode resultar.

Esta possibilidade de pensar a arte e a tecnologia não nos leva obrigatoriamente a pensar na Arte Digi-

tal, enquanto arte que se faz pelo meio digital. Acreditamos muito que é possível envolver suporte tecnológico na criação artística mesmo que o computador não seja o protagonista ou o centro da atenção do projeto artístico. Além disso, pensamos em uma arte participativa, colaborativa, performática e que pode ser nutrida por aspectos computacionais mas não necessariamente se dá pelo digital. Esta arte é acolhida no ALICE por meio de Orquídea, nossa Orquestra de Ideias.

Isso tudo era mais simples quando nossos encontros aconteciam de forma presencial e nossos pesquisadores e pesquisadoras podiam trocar experiências sentados e sentadas em cadeiras. A situação do mundo pós Março de 2020 fez com que nosso coletivo se espalhasse por Minas Gerais e estados vizinhos, o que nos levou a procurar e tentar encontrar outras formas de cooperação (ver Mapa 2¹). Também nos levou a entender nosso espaço como outro espaço, etéreo e espalhado, com diferentes tempos e rotinas. E é diante deste cenário que apresentamos as pesquisas que estão sendo feitas no ALICE nos anos de pandemia.

¹Disponível em : https://umap.openstreetmap.fr/en/map/alice_635354#7/-20.267/-41.616.



Figura 1: Criações da Orquídea. Arte: Mariana Lellis

tradição e sua religião. Em algumas das reações, foi perguntado quem autorizou a produção. Tal pergunta subentende a existência de uma espécie de “donos” dos sons, que poderia autorizar ou não a releitura. Porque as pessoas se preocupam com a posse de um som, que arbitrariamente preenche a paisagem sonora do centro urbano? Se os sons não pedem permissão para tocar, o artista deveria pedir permissão para dialogar com eles? Outra reação que chama a atenção é o aparente incômodo de alguns com relação à africanidade percussiva dos sons, realçada pela adição de instrumentos e batidas. Qual é a origem desse estranhamento [5]?

2.3. A persona do músico na pandemia: era digital e relação público artista nas redes sociais.

A produção musical no século XXI vem transformando a maneira dos músicos criarem e apresentarem suas obras. No início de 2020, com o surgimento da pandemia COVID-19 e o distanciamento social decorrente, todo setor da cultura na área de apresentações ao vivo foi interrompido, deixando desempregados muitos trabalhadores da cultura. Surgiu, então, novas oportunidades de escoamento da produção artística a partir das redes sociais. A atuação dos músicos neste novo cenário nos permite pensar que o próprio paradigma da arte sofreu mudanças epistêmicas radicais no que diz respeito à relação entre sujeito e objeto, devido à proximidade do público com o artista que este formato permite, mudando o foco da intenção artística da obra em si para o processo da experiência artística. Assim, entendemos que há em curso a transformação da aparência social a que os músicos e musicistas estão vinculados no seu trabalho. Se nos meios eruditos europeus o músico era visto ou idealizado como virtuoso, no contexto norte-americano, que se espalha pelo Brasil, dentro da cultura pop, como celebridade, e na cultura popular do início do século XX como malandro e boêmio, como está sendo construída, então, essa persona hoje? [6]

2.4. A brasilidade como inspiração para o design de superfícies

O tema da pesquisa é a relação da ‘brasilidade’ com o design e a arte digital, estudando a representação da ‘brasilidade’ em diferentes campos do design, abrangendo principalmente o design de superfícies e o design de moda. Busca-se também discutir e aplicar conceitos-chave da arte digital ao longo do desenvolvimento dos produtos e da pesquisa em si. A problematização do trabalho parte da seguinte pergunta: seria a cultura brasileira amplamente explorada no campo do design como elemento de inspiração para o desenvolvimento de estampas e motivos, principalmente no que concerne ao campo do design de moda?

Portanto, pode-se dizer que o objeto de pesquisa são os elementos da ‘brasilidade’ presentes na arte nacional como inspiração para o desenvolvimento de estampas. Dentro da pesquisa, entende-se como “elementos da brasilidade” as partes integrantes da cultura brasileira, ou seja, características que a tipificam e a compõem. A pesquisa busca abordar temas como sustentabilidade, design digital, moda contemporânea, entre outros, estabelecendo relações

entre as dimensões comuns ao sistema de moda e à arte digital, levantando ainda discussões sobre o uso de softwares open-source e metodologias baseadas na prática artística.

2.5. Colaboração em arte digital

A colaboração pode existir de diferentes formas e em diferentes etapas de um projeto, envolvendo seres humanos e computadores. O Modelo 3C discute a colaboração considerando três pilares, necessários para que a mesma ocorra: a comunicação, a cooperação e a coordenação [7].

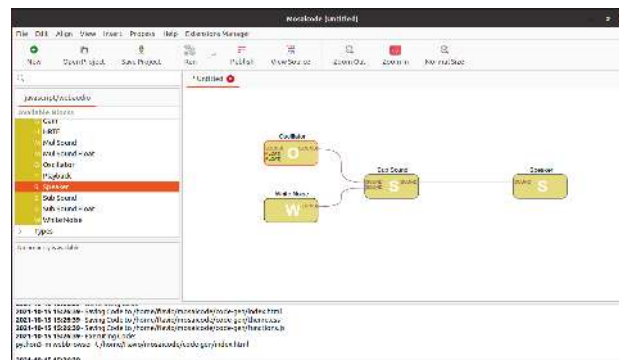


Figura 4: Tela do Mosaicode apresentando uma criação para webaudio

O computador tem sido cada vez mais utilizado para apoiar a colaboração, principalmente após o início da pandemia de COVID-19 – a tela é o que nos conecta agora. O trabalho remoto veio de forma brusca neste momento, com o desafio de adaptação para essa maneira de trabalhar, dificultando os trabalhos artísticos e multidisciplinares.

As pesquisas realizadas no laboratório ALICE durante a pandemia foram feitas remotamente, com o auxílio do computador. Diferentes tipos de aplicações e tecnologias são utilizadas no trabalho remoto para coordenar, cooperar e comunicar, durante o processo de colaboração. Desde o funcionamento de redes de computadores, hardwares e softwares, aos processos colaborativos envolvidos.

Os processos colaborativos podem ser diferentes dependendo do contexto que está sendo aplicado, divergindo também as aplicações e tecnologias de apoio utilizadas. Equipes de desenvolvimento de software costumam utilizar ferramentas de controle de versão e canais de comunicação como apoio ao processo criativo presente no desenvolvimento de software.

No contexto de arte digital, o apoio do computador pode acontecer durante o processo criativo da obra ou durante a performance. A computação oferece recursos que apoiam a arte digital, como: o processamento, a síntese e a análise de som e imagem; a interação humano-computador através de sensores, atuadores, GUIs e interfaces comuns (controladores MIDI, joysticks...); e a comunicação remota via redes de computadores [8].

Os diferentes contextos e áreas envolvidos em processos colaborativos demandam compartilhamento (envio e recebimento) de diferentes artefatos de software. Os

dados compartilhados podem abstrair diferentes mídias digitais e devem ser coordenados para que apenas os usuários com permissão tenham acesso. Além de compartilhar, é importantes ter o controle de versão dos artefatos, forma de empacotamento de distribuição, informando os autores, a licença e datas de criação e modificações [9].

Pensando no contexto de arte digital e seus processos criativos, surgiu no laboratório ALICE a ferramenta Mosaiccode, apresentada na Figura 4. Essa ferramenta tem como propósito o apoio à criação de arte digital, mas não se limita a esse domínio. O Mosaiccode tem como intuito diminuir o esforço necessário para desenvolvimento de aplicações a partir das vantagens presentes no paradigma de programação visual (VPL – Visual Programming Languages) para domínio específico (DSL – Domain Specific Languages) [10].

2.6. Sunflower: um protocolo para padronização de ambientes de Internet das Coisas Musicais (IoMusT)

Internet das Coisas Musicais (IoMusT) é um campo de atuação que surgiu a partir da expansão dos domínios da Internet das Coisas (IoT) para a prática musical. A partir disso, surgem correlações com diversas áreas, como interação humano-computador (IHC), novas interfaces para expressão musical, jogos, arte participatória, música ubíqua e tantas outras, caracterizando-o como um setor interdisciplinar [11].

Apesar de já ser bem-estruturado e propor novas formas para apresentações musicais, esse campo também lida com uma gama de problemas, indo desde os artísticos até aqueles socioeconômicos e ambientais. De um ponto de vista computacional, os maiores problemas residem na falta de padronização e interoperabilidade entre os objetos presentes neste ambiente [11, 12].

Neste contexto, um dos trabalhos desenvolvidos em nosso laboratório é o Sunflower[13, 14], um protocolo desenvolvido para atenuar os problemas de padronização de dados, protocolos e objetos. Seu modo de funcionamento é análogo a arquitetura Pipes-and-Filters, onde as coisas musicais irão se comportar como filtros, ou seja, unidades independentes que não conhecem as propriedades de seus vizinhos e comunicam-se apenas a partir de seus dados de entrada e saída. Já os dutos podem ser observados nos meios de transmissão pela rede, que somente transportam dados, não aplicando processamento a eles.

Para validar a funcionalidade deste protocolo, foi proposto um ambiente dividido em quatro camadas: áudio, gráfica, controle e gerenciamento, sendo classificadas de acordo com os dados, objetos, protocolos e atores presentes em cada uma delas, como apresentado na Figura 5. Desse modo, os autores puderam realizar testes acerca do formato das mensagens trocadas, o comportamento das coisas musicais e também questões relacionadas a rede, como latência e *jitter*. As principais contribuições do Sunflower residem justamente no seu protocolo de uso, em especial no modo de funcionamento semelhante a arquitetura

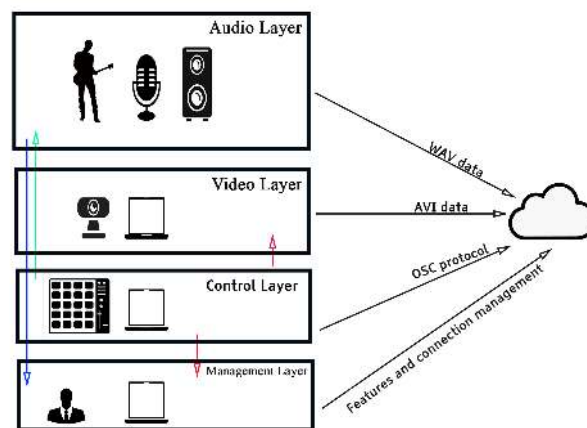


Figura 5: Camadas do ambiente sunflower

de *software* mencionada. A partir dessa pesquisa, os autores não pretendem encerrar a discussão de como devem ser os ambientes de IoMusT, tampouco apontam quais ferramentas devem ou não ser utilizadas, buscando apenas contribuir para a área ao propor uma nova ferramenta.

2.7. Telefone sem fio

Pensando nos fenômenos das comunicações em massa e nos efeitos que a reprodução de uma mensagem pode criar, apresentamos aqui o trabalho artístico chamado Telefone sem fio, uma composição para celulares em rede que utilizam o ambiente para criar camadas sonoras[15]. Os telefones ainda utilizam suas telas, lanternas e vibracall para auxiliar o público a compreender a comunicação nos dias atuais. A peça conta ainda com um conjunto de amplificadores acústicos feitos em cerâmica que servem para amplificar as vozes dos celulares e nos ajudam a compor a cenografia desta apresentação, como apresentado na Figura 6.



Figura 6: Amplificadores cerâmicos e celulares na peça Telefone sem fio

2.8. Surfing with Narcissus

A peça Narcissus de Thea Musgrave foi composta para clarineta ou flauta e delay na década de 80 e é uma peça que depende, para sua execução, de determinado aparato tecnológico que não se encontra mais disponível no mercado. A tentativa de recriar a peça utilizando o computador como aparato tecnológico motivou a parceria entre pesquisadores do Departamento de Música, o professor e clarinetista

Dr. Iura de Rezende e do Departamento de Computação da Universidade Federal de São João del-Rei e resultou na presente pesquisa. Tal pesquisa trouxe à tona diversas questões sobre a montagem desta peça e da performance da mesma utilizando, para isso, uma ferramenta criada para a web para recriar a peça, como mostra a Figura 7 [16, 17].

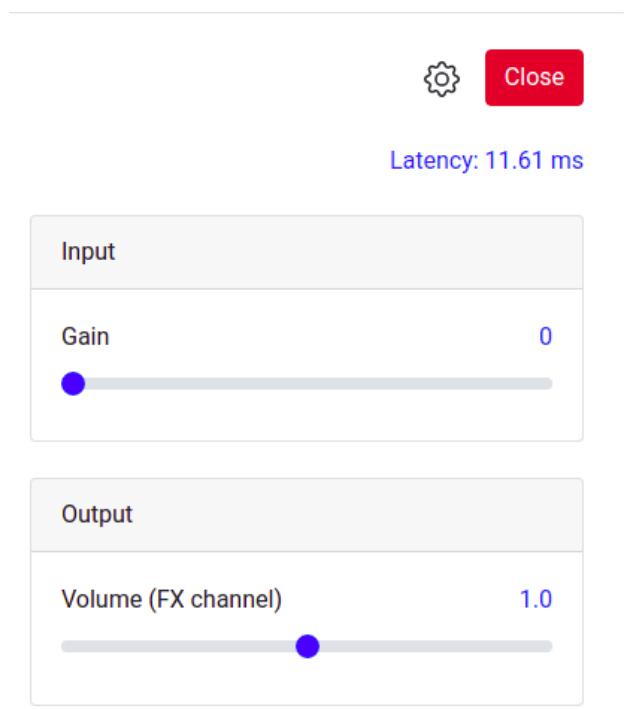


Figura 7: Tela de controle para a aplicação Web desenvolvida para a criação da peça Narcissus

2.9. Gluetube e a colagem de material online

A colagem é uma técnica muito usada na criação de arte que consiste em literalmente recortar e agrupar partes de diversos elementos existentes para obter um novo resultado artístico. Mais do que uma técnica, a colagem é uma linguagem que traz também a discussão de direito autoral ao imaginar que a utilização de parte de uma peça para criar outra poderia ser uma violação de direitos. Com o avanço da Internet, a rede passou a ser o local onde obras de arte são divulgadas tornando-se assim, um espaço de possibilidades infinitas de colagem. Visando possibilitar a colagem de conteúdo online, este pesquisa apresenta a ferramenta GlueTube para colagem de material online e suas possibilidades para a criação musical [18, 19].

3. Discussões

Antes a pesquisa sobre a colaboração em processos criativos mediada pela tecnologia ou a pesquisa da Internet das Coisas Musicais levava em conta a sincronização de relógios com milissegundos para permitir que pessoas pudessem tocar juntas pela rede. Desde o início destas pesquisas muita coisa mudou. Hoje pensamos em outras sincronizações para trabalharmos juntos como o tempo da vacina em cada cidade ou a idade de cada pessoa de nosso grupo na fila para vacinar. A latência da vacina hoje nos

preocupa mais do que o os pacotes de rede em TCP demorarem mais para serem entregues do que os pacotes UDP. Neste momento temos alunos que já tomaram a segunda dose da vacina e outros que estão aguardando a primeira. Estes são os buffers que tem nos preocupado.

Mesmo assim, a rede segue sendo nossa aliada para a colaboração e a cooperação. De alguma forma, aprendemos a trabalhar desta maneira e transformamos nossas dificuldades em questões de pesquisa para nos ajudar a entender as possíveis fronteiras de pesquisa em Arte, Cultura e Sustentabilidade Digital [20]

4. Agradecimentos

Agradecemos ao CNPq e a PROPE / PROEX / PROAE da UFSJ, por financiar toda a pesquisa e também, a todos os membros do laboratório de pesquisa, ALICE, que contribuíram para estes trabalhos.

Referências

- [1] João Teixeira Araújo. *Proposta de Arquitetura para Interação do Público em Performances Digitais*. Dissertação (mestrado em ciência da computação), Universidade Federal de São João del-Rei, 2021.
- [2] Fábio dos Passos Carvalho. *Paisagens Sonoras: O toque dos sinos na era digital*. Dissertação (mestrado em artes, urbanidades e sustentabilidade), Universidade Federal de São João del-Rei, 2021.
- [3] Igino de Oliveira Silva Júnior. *PROJETO RENDA-SE: A APROPRIAÇÃO CRIATIVA DE ESPAÇOS URBANOS PROMOVIDA POR INTERFACES LÚDICAS*. Dissertação (mestrado em artes, urbanidades e sustentabilidade), Universidade Federal de São João del-Rei, 2021.
- [4] João Teixeira Araújo, Igino de Oliveira Silva Junior, Rômulo Augusto Vieira Costa, Marcela Alves de Almeida, and Flávio Luiz Schiavoni. Using game theory to support the audience participation in digital performances. *Per-Musi*, 1(40):1–13, 2021.
- [5] Samuel Rodrigues Rabay and Flávio Luiz Schiavoni. Sampleando sinos afrocatólicos - territorialidade, conservadorismo e antropofagia. In *Caderno de Resumos: Sonoridades fronteiriças - II Conferência Internacional de Pesquisa em Sonoridades*, pages 96–97, 2021.
- [6] João Marcos de Oliveira Lara, Zandra Coelho Miranda, and Flávio Luiz Schiavoni. A persona do músico na pandemia: era digital e relação público artista nas redes sociais. In *Caderno de Resumos: Sonoridades fronteiriças - II Conferência Internacional de Pesquisa em Sonoridades*, pages 192–193, 2021.
- [7] Hugo Fuks, Alberto Barbosa Raposo, Marco Aurélio Gerosa, and Carlos José Pereira de Lucena. *Applying the 3C model to groupware engineering*. PUC Rio de Janeiro, 2004.
- [8] Flávio Luiz Schiavoni and Luan Luiz Gonçalves. From virtual reality to digital arts with mosaiccode. In *2017 19th Symposium on Virtual and Augmented Reality (SVR)*, pages 200–206, 2017.
- [9] José Mauro da Silva Sandy. *UAISharing - Interface Universal de Acesso a Recursos Compartilhados*. Dissertação (mestrado em ciência da computação), Universidade Federal de São João del-Rei, 2019.

- [10] Flávio Luiz Schiavoni, Luan Luiz Gonçalves, and André Lucas Nascimento Gomes. Web audio application development with mosaiccode. In *Proceedings of the 16th Brazilian Symposium on Computer Music*, pages 107–114, São Paulo - SP - Brazil, 2017.
- [11] Luca Turchet, Carlo Fischione, Georg Essl, Damián Keller, and Mathieu Barthet. Internet of musical things: Vision and challenges. *IEEE Access*, 6:61994–62017, 2018.
- [12] Rômulo Vieira and Flávio Luiz Schiavoni. Can pipe-and-filters architecture help creativity in music? In *Proceedings of the Workshop on Ubiquitous Music 2020*, pages 109–120, Porto Seguro, BA, Brasil, November 2020. Zenodo.
- [13] Rômulo Vieira and Flávio Luiz Schiavoni. Managing an iomust environment and devices. In *Proceedings of the Workshop on Ubiquitous Music 2021*, 09 2021. in press.
- [14] Rômulo Vieira and Flávio Luiz Schiavoni. Sunflower: an environment for standardized communication of iomust. In *Proceedings of the Audio Mostly 2021*, 09 2021. in press.
- [15] Carlos Eduardo Oliveira de Souza, Jônatas Araújo da Silva, and Flávio Luiz Schiavoni. Telefone sem fio. In *Anais do Ubimus 2021*, page 4, Porto, 2021.
- [16] Luan Luiz Gonçalves, Rafael Alves Soares de Andrade, Iura de Rezende Ferreira Sobrinho, and Flávio Luiz Schiavoni. Surfing with narcissus: Updating thea musgrave’s work for solo clarinet. In *Proceedings of the Web Audio Conference (WAC-2021)*, pages 1–5, 2021.
- [17] Iura de Rezende, Rafael Alves Soares de Andrade, and Flávio Luiz Schiavoni. Lições aprendidas com a peça narcissus de thea musgrave. In *Anais do XXX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 1–11, Manaus, AM, Brasil, December 2020. ANPPOM.
- [18] Avner Maximiliano de Paulo, Carlos Eduardo Oliveira de Souza, and Flávio Luiz Schiavoni. Gluetube e a colagem de conteúdo online. In *Anais do XXX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, pages 1–100, Manaus, AM, Brasil, December 2020. ANPPOM.
- [19] Avner Maximiliano de Paulo, Carlos Eduardo Oliveira de Souza, and Flávio Luiz Schiavoni. Gluetube: Música concreta com o youtube. In *Anais do Workshop de Computação Teórica e Aplicada*, page 23, São João del-Rei - MG, 2019.
- [20] Flávio Luiz Schiavoni. Possíveis fronteiras de pesquisa em arte, cultura e sustentabilidade digital. In *Anais do I Seminário do Grupo Transdisciplinar de Pesquisa em Arte e Sustentabilidade (GTrans) da Universidade Federal de São João del - Rei (UFSJ)*, pages 44–55, 2021.

CEGeME 2021 - Gestos, Movimentos, Performance musical e Pandemia

Maurício Alves Loureiro, Alexandre Silva, Aluizio Barbosa de Oliveira Neto,
Davi Alves Mota, Felipe Brandão Barros, Flávio Luiz Schiavoni,
Gustavo Machado Oliveira, Luis Fernando Umbelino da Silva, Ravi Shankar Viana Domingues,
Renato Rodrigues Lisboa, Tairone Nunes Magalhães,
Thiago de Almeida Magalhães Campolina, Tiago Lima Bicalho Cruz¹

¹ CEGeME - Centro de Estudos do Gesto Musical e Expressão
Escola de Música
UFMG - Universidade Federal de Minas Gerais

<https://musica.ufmg.br/cegeme/>

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, is dedicated to 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. 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 group has attracted study postulants from different fields and professional musicians instigated by specific inquiries on the performance on their instruments, such as studies on the use of the modern bass trombone second valve for achieving a better legato, the source-filter acoustic parameters in the trumpet, the facial emotions present in the performance of lyrical singers, the acoustic and psychoacoustic parameters associated with different reed preparations for the oboe, the spectral modeling of note attack on the clarinet, musical timbre analysis based on machine learning, studies of acoustics and kinematics parameters associated to the execution of legato note transition on the clarinet, analysis of note articulation on the performance of a clarinet piece by Debussy, sonological applications in musical education, the influence of musical interpretation consistency on the cohesion of a chamber music performance and the software development on python platform created to support the research demands of the research conducted at CEGeME.

Resumo. O CEGeME - Centro de Pesquisa em Gesto e Expressão Musical é filiado ao Programa de Pós-Graduação em Música da Universidade Federal de Minas Gerais (UFMG), sediado na Escola de Música de Belo Horizonte, Brasil, desde 2008 é dedicado à pesquisa empírica da performance musical, que partem de informação de conteúdo musical extraída de sinais de áudio e da posição espacial tridimensional dos músicos, registradas durante uma performance musical. Com o objetivo de estabelecer uma abordagem musicológica a diferentes aspectos da expressividade musical, investigamos relações causais entre a intenção expressiva dos músicos e a forma como manipulam o material acústico e como se movem durante a execução

de uma peça musical. O grupo atraiu postulantes de estudos de diversas áreas de conhecimento e músicos profissionais instigados por questões específicas envolvidas na expressividade musical, tais como, estudos sobre o uso da segunda válvula do trombone baixo moderno para obter um melhor legato, os parâmetros acústicos do filtro-fonte no trompete, as emoções faciais presentes na performance de cantores líricos, os parâmetros acústicos e psicoacústicos associados a diferentes preparações de palheta do oboé, a modelagem espectral do ataque de notas na clarineta, análise do timbre musical baseada em aprendizado de máquina, estudos de parâmetros acústicos e cinemáticos associados à execução da transição de notas em legato na clarineta, análise da articulação na performance de uma peça para clarineta de Debussy, aplicações sonológicas na educação musical, influência da consistência na interpretação na coesão da performance da música de câmara e o desenvolvimento de software na plataforma python criado para dar suporte às demandas da pesquisa conduzida no CEGeME.

1. Introdução

O grupo de pesquisa CEGeME – Centro de estudos do Gesto Musical e Expressão (UFMG) realiza projetos de pesquisa empírica da performance musical, desde sua criação em 2008. A pesquisa do CEGeME parte de informação de conteúdo musical recuperado do sinal de áudio e da posição espacial tridimensional do músico, gravados durante a execução de um excerto musical. Poucos anos antes de sua criação, conduzimos nossos primeiros estudos neste campo no CEFALA (Centro de Pesquisas da Fala, Acústica, Linguagem e Música), um grupo de pesquisa localizado na Escola de Engenharia da UFMG, coordenado pelo professor Hani Yehia. Estes estudos investigaram o timbre musical a partir de um modelo computacional para representar variações dinâmicas do timbre da clarineta [1, 2, 3].

Os resultados deste estudo motivaram Mauricio Loureiro, atual coordenador do CEGeME, a buscar ferramentas e modelos computacionais para abordar outros problemas da performance musical, o que resultou na criação de um grupo interdisciplinar voltado para a pesquisa empírica em performance musical. O CEGeME atraiu postulantes de estudos de diversas especialidades, como Ciência da Computação, Engenharias, Física, Fonoaudiologia e Musicoterapia, além de colaborações de músicos profissionais

instigados por indagações específicas sobre a execução de seus instrumentos. O grupo conta atualmente com a participação de alunos de graduação, mestrado, doutorado e pesquisadores visitantes em estágio de pós doutorado [4].

Vinculados ao Programa de Pós-Graduação em Música da Universidade Federal de Minas Gerais (UFMG), sediado pela Escola de Música da UFMG, em Belo Horizonte, Minas Gerais, Brasil, os laboratórios do CEGeME contam com instalações e equipamentos adequados para aquisição e análise de dados acústicos e de movimento corporal de músicos, os quais têm sido patrocinados por bolsas e auxílios financeiros de agências brasileiras de fomento à pesquisa (CNPq, FAPEMIG e CAPES). Além do CEFALA, o CEGeME tem ainda parceria com outro grupo de pesquisa da Escola de Música, o LAPIS – Laboratório de Performance com Sistemas Interativo, coordenado por Sérgio Freire Garcia.

Com foco em diferentes aspectos da expressividade musical, propomos modelos analíticos que buscam identificar relações causais entre a intenção expressiva dos músicos e a maneira como eles manipulam o material acústico e como se movem ao interpretar uma obra musical. A metodologia busca suporte em conhecimentos como modelagem computacional, análise estatística e processamento digital de sinais, que agregam habilidades à musicologia tradicional. Os inúmeros projetos concluídos e em andamento desenvolvidos pela equipe do CEGeME, são norteados por 4 eixos que abordam diferentes problemas envolvidos neste campo de pesquisa, às vezes por vias distintas, adotando novos protocolos de experimentos e novas técnicas de análise, visando sempre aperfeiçoar nossos modelos metodológicos:

1. **Extração multimodal de informação musical:** Consiste no desenvolvimento e melhoria contínua de ferramentas, algoritmos e modelos computacionais para extração de parâmetros acústicos e cinemáticos do sinal de áudio e da posição espacial tridimensional dos músicos, que possam encapsular informação de conteúdo musical. É uma modalidade de pesquisa em constante desenvolvimento e aperfeiçoamento visando dar suporte aos projetos desenvolvidos no CEGeME.
2. **Estudo empírico da música tocada:** Com foco em diferentes aspectos da expressividade musical, esses projetos propõem modelos analíticos que buscam identificar relações causais entre a intenção expressiva dos músicos e a forma como manipulam o material acústico e como se movem na interpretação de uma obra musical.
3. **Interação e coordenação em performance musical de conjuntos:** Com o objetivo de investigar o processo de sincronização / coordenação em pequenos conjuntos musicais, os projetos desta categoria buscam modelar as interações dos músicos em situações típicas de performance, a partir de respostas visuais e acústicas coletadas durante simulações de performances musicais reais. Padrões de sincronização temporal e outros padrões

de acoplamento, manifestados por meio de manipulações sonoras e movimentos corporais, são observados e analisados, visando mapear elementos acústicos e gestuais que possam influenciar no ajustamento entre os músicos.

4. **Aplicações sonológicas na aprendizagem musical e desenvolvimento humano:** Projetos que têm por objeto de investigação o comportamento humano em atividades musicais mediadas por ferramentas tecnológicas voltadas para a aprendizagem e desenvolvimento humano. Com este tipo de estudo, o CEGeME pretende ampliar o escopo de sua pesquisa, visando verificar o potencial de aplicação que os resultados e a tecnologia gerada por estes estudos possam oferecer a outros campos de pesquisa, tais como a pedagogia do instrumento musical e a musicoterapia.

2. Projetos do CEGeME

Os seguintes projetos foram desenvolvidos recentemente ou estão sendo desenvolvidos atualmente no CEGeME.

2.1. A eficiência na produção do legato utilizando o segundo rotor isoladamente do trombone baixo

O Trombone Baixo é um instrumento que chegou a seu estágio atual de desenvolvimento na década de 1950. O instrumento passou por alterações ao longo dos séculos, tais como como afinações distintas, alterações nas dimensões e calibre do tubo e da campana, as quais incluiu também a incorporação de duas válvulas visando facilitar as dificuldades impostas pelo aumento das dimensões do instrumento. Apesar de já consolidado no cenário musical profissional, há ainda uma carência de estudos sobre técnicas específicas para a utilização dessas válvulas, denominadas de rotores. No trombone baixo, além das sete posições básicas da vara do trombone tenor, o instrumentista se depara ainda com seis posições para cada um dos rotores, mais cinco com os dois rotores acionados [5]. Por ter sido relativamente recente a adição destas válvulas, a extensão das possibilidades do trombone baixo com o uso de seus dois rotores, quer combinados ou isoladamente ainda não é devidamente conhecida, principalmente por estudantes dedicados a este instrumento. A pesquisa de doutorado em andamento tem como objetivo principal verificar a eficiência da execução do legato no trombone baixo utilizando o 2º rotor isoladamente através de uma abordagem quantitativa, contribuindo dessa forma com uma pesquisa original para trombonistas baixo e professores de trombone de modo a facilitar a compreensão do funcionamento e uso dos rotores do instrumento.

Foram estabelecidas 5 condições de utilização do segundo rotor: (1) posições básicas sem utilização do segundo rotor ; (2) acionando somente o segundo rotor; (3) acionando somente o primeiro rotor; (4) acionando ambos os rotores; (5) posições alternativas sem uso de rotor. Três trombonistas baixo profissionais foram recrutados para executar excertos selecionados da obra sinfônica de Anton Bruckner, Richard Wagner e Johannes Brahms

que contêm transições de notas em legato adequadas para a sua execução em duas ou três destas condições. Ambos os excertos de Bruckner foram executados nas condições 1 e 2, o excerto de Wagner nas condições 1, 2 e 3 e excerto de Brahms nas condições 1, 2 e 5. Foram registradas 6 execuções de cada excerto em cada uma dessas condições, totalizando 180 gravações.

A segmentação de notas foi estimada utilizando o software Sonic Visualizer, auxiliado pelas curvas de fluxo espectral e energia RMS. A diferença de tempo gasto na execução das transições entre as diversas condições foi estimada e testada. A duração da transição de nota foi definida como o intervalo de tempo entre o instante de início da queda da energia da nota e o final do ataque da nota seguinte. A diferença de duração entre as condições 1 (somente segundo rotor) 2 (sem utilizar o rotor, etc..) foi comparada aplicando um teste t pareado para evitar qualquer influência de viés de três outros fatores considerados no experimento: características de transição, músico e tomada. Resultados preliminares mostram uma queda significativa na duração das transições na condição 1, indicando uma possível incremento da eficiência na produção do legato com a utilização do segundo rotor [6].

2.2. Análise do pulso labial no trompete por filtragem inversa

O controle do sopro na execução do trompete é um elemento essencial da técnica do instrumento, principalmente na execução de notas de uma mesma posição. Este estudo buscou informação relacionada a esse controle no pulso de ar que passa pelos lábios em vibração. A medição direta deste pulso é complexa e invasiva por envolver medições dentro do instrumento e dentro da boca do trompetista.

Assumindo a produção sonora do trompete como resultante do sinal de excitação gerado pela vibração dos lábios, filtrado pelas ressonâncias do tubo do instrumento, o modelo fonte-filtro adotado em estudos da produção de voz, no qual a fonte é sinal de excitação gerado na glote e o filtro as ressonâncias do trato vocal, foi proposto para a estimação do pulso de ar que passa pelos lábios do trompetista. Utilizando metodologia análoga para estimar o fluxo glotal em estudos da produção da voz, este estudo propôs estimar o pulso labial na produção sonora do trompete pela filtragem inversa do som produzido pelo instrumento captado por microfone [7].

A variação da forma do pulso assim estimado mostrou forte correlação positiva da obliquidade (skewness) com a altura da nota tocada, que pode indicar que a abertura dos lábios em notas agudas executadas no trompete ocorre de forma mais abrupta do que em notas graves. O método, se devidamente validado pode mostrar perspectivas de contribuição para a obtenção de informações valiosas sobre a técnica de execução do trompete [7, 8, 9].

2.3. Emoções no canto e expressões faciais

Entendendo que o cantor de ópera também é um ator que interpreta o texto da ária ou recitativo, cada letra dessas peças contém mensagens que despertam algum tipo de emo-

ção. O cantor, além de cantar as notas musicais, interpreta o texto para transmitir a emoção contida na letra. Na fala, as emoções influenciam parâmetros como frequência fundamental, intensidade, ritmo, e andamento. No canto, embora o cantor esteja teoricamente limitado ao que está escrito na partitura, diferentes emoções podem modificar os mesmos parâmetros da fala, mesmo em graus mais discretos. No entanto, ao contrário da fala, em que as expressões faciais tendem a ser espontâneas e naturais, no canto tais emoções são pensadas e fabricadas pelo cantor.

Neste estudo, buscamos verificar se existe um padrão de emissão vocal e expressão facial na interpretação de diferentes emoções no canto. O objetivo desta pesquisa é verificar como as diferentes emoções básicas influenciam a voz e as expressões faciais do cantor lírico e quais parâmetros de expressividade podem ser verificados durante o canto [10].

Para a avaliação dos cantores, as tarefas serão selecionadas dentro de um contexto musical e, para tanto, uma pequena seção musical foi composta com texto neutro e que, a princípio, não desperta nenhuma emoção pelo seu significado. As tarefas são compostas por intervalo de salto, movimentos melódicos ascendentes e descendentes e um crescendo em uma nota sustentada. O excerto será cantado em diferentes condições de emoções básicas: felicidade, tristeza, medo, raiva. Além disso, será solicitado ao cantor que cante de forma neutra. As tarefas serão gravadas em áudio para análise acústica e também serão coletados outros dados da performance como os dados do EGG, o vídeo frontal dos cantores e a posição espacial tridimensional de diversos pontos da face capturados por meio de um equipamento de alta precisão (Qualysis Pro-reflex). Cantores líricos, sopranos e tenores, com idades entre 25 e 45 anos, serão avaliados. A possibilidade de coleta de dados de cantores líricos com idade entre 18 e 25 anos, sem experiência em ópera, também é estudada para comparar com o grupo de profissionais e verificar se a experiência tem algum tipo de influência positiva na interpretação do texto cantado.

2.4. Análise de parâmetros acústicos e psicoacústicos da sonoridade do oboé associados aos diferentes estilos de raspados de palheta

O oboé é um instrumento de sopro da família das madeiras que utiliza uma palheta dupla. Sua produção sonora ocorre pela vibração de duas lâminas flexíveis de um tipo de bambu, *Arundo Donax*, que se contrapõem e formam um duto para a passagem do ar, possibilitando a entrada do fluxo do ar modulado por sua própria vibração [11, p.35]. Grande parte das manipulações dos parâmetros musicais, tais como timbre, fraseado, articulação, dinâmica, afinação e vibrato, utilizados para a construção de uma interpretação musical, são realizadas através da sua interação do oboísta com a sua palheta. Além de aspectos técnicos e expressivos, desenvolvidos ao longo da trajetória musical de todo instrumentista, a aprendizagem do processo de produção da palheta é fundamental na formação do oboísta [12, p. 62].

Atualmente, no Brasil e no mundo, há duas escolas de raspado de palhetas de oboé que prevalecem sobre as demais: a escola americana (raspado longo) e escola alemã/europeia (raspado curto). Independentemente do raspado utilizado, os oboístas buscam uma palheta que lhes permitam manipular adequadamente estes parâmetros para realizar com maior eficiência possível suas intenções expressivas em diferentes contextos musicais e melhor adaptar às diferentes situações de performance. Saber fazer a própria palheta é imprescindível para a maioria dos oboístas profissionais, pois, o estilo interpretativo de um oboísta e a realização da sua sonoridade ideal são determinados em grande parte pela palheta escolhida, que afetará diretamente no desenvolvimento técnico e interpretativo do instrumentista[13]. Por essa razão, a produção da palheta, bem como seu processo de aprendizagem, ocupa parte da prática do oboísta, sendo um aspecto fundamental em seu processo formativo [12, p. 58].

Entre 2015 e 2018 realizamos uma pesquisa cujo objetivo foi analisar parâmetros acústicos e psicoacústicos relacionados à sonoridade e à articulação de oboístas que utilizam os dois principais modelos de raspagem de palhetas de oboé, correlacionando-os com os conceitos subjetivos utilizados para descrição desses parâmetros musicais buscando compreender a influência dos estilos de raspado e sua relação com a cultura sonora do oboísta. Realizamos um estudo descritivo do tipo transversal entre setembro e dezembro de 2017, cuja coleta de dados foi dividida em duas etapas. Primeiramente foi aplicado um questionário, semi-estruturado, a 21 oboístas profissionais, brasileiros e estrangeiros, que gravaram dois excertos contrastantes, extraídos da parte de oboé de obras sinfônicas. Em seguida foi realizado um teste subjetivo de percepção em 73 músicos, docentes e discentes de três universidades brasileiras.

Com a utilização do EXPAN[14], seis descritores acústicos foram extraídos do sinal de áudio da gravação dos excertos (centróide espectral, achatamento espectral, irregularidade espectral, duração do ataque, índice de articulação e índice de legato), que foram confrontados com os dados coletados do questionário e do teste subjetivo referentes ao timbre e a articulação. Resultados revelaram que os dois raspados de palheta testados podem influenciar a sonoridade e articulação. Foi observado também que a maneira com que os parâmetros acústicos são manipulados, direcionada pelas referências e as demandas do contexto musical, influencia o processo de produção e os ajustes das palhetas de oboé.

2.5. Caracterização de Ataques na Clarineta via Modelagem e Decomposição Espectral

O ataque das notas pode assumir papéis importantes em performances musicais. No ataque, músicos são capazes de controlar timbre, articulação e ritmo, manipulações essenciais para transmitir suas intenções expressivas e tornar a música mais apreciável. Na literatura, ao mesmo tempo que a importância dos ataques para a percepção e construção da performance é evidenciada, são expostas também as dificuldades de analisá-los. Eles são eventos acústicos

de curta duração, sujeitos a interações com a reverberação do ambiente, e com características transientes e não lineares. A modelagem desses eventos busca um melhor entendimento sobre como são manipulados os parâmetros relacionados à expressividade.

Uma tese concluída recentemente [15] com o suporte do CEGeME propõe a modelagem e decomposição espectral como uma via para a caracterização de ataques na clarineta. A primeira parte dos resultados (publicada em [16]) confronta em duas dimensões (2D), proporções de energia (em relação ao sinal original) de três componentes espectrais extraídas no ataque. Nessa representação 2D (nomeada de plano **H2R2**), foi possível verificar visualmente e fazer inferências sobre as proporções das componentes harmônicas, não-harmônicas e interações com a acústica da sala, visando enriquecer o estudo das performances musicais na condição de prática cotidiana, onde ambientes reverberantes estão sempre presentes. Na segunda parte dos resultados (publicada em [17]), foi proposto um descritor tridimensional, nomeado de Perfil Harmônico do Ataque (**PHA**), envolvendo tempo, frequência e magnitude espectral, com o objetivo de caracterizar a estrutura harmônica de ataques na clarineta. O **PHA** foi definido como a sequência de inícios dos harmônicos, começando pela frequência fundamental, seguindo a ordem harmônica crescente. O início de cada harmônico foi considerado como 50% da faixa de variação de sua magnitude em dB no trecho do ataque.

Para o plano **H2R2** e o **PHA**, foram examinados ataques de notas em transições com diferentes demandas técnicas de execução, em condição de legato e articuladas. Na análise do plano **H2R2**, testes estatísticos para comparação entre médias bidimensionais (MANOVA - Multivariate Analysis of Variance) indicaram diferenças significativas ($p < 0,05$), considerando “músico” como fator. Para o **PHA**, testes MANOVA utilizando redução supervisionada de dimensionalidade dos dados através de Análise Discriminante Linear também indicaram separação considerando o fator “músico”.

2.6. Modelo de análise de timbre em performances musicais utilizando aprendizado de máquina

A área de performance musical reconhece que instrumentistas utilizam diversas manipulações do instrumento para alcançar diferentes variações do som e, com isto, alcançar uma maior expressividade na performance musical. Tal expressividade difere da expressividade proposta pelo compositor já que o compositor, ao notar a peça em uma partitura, costuma se limitar a definir alguns parâmetros do som como intensidade, duração e altura. Além disso, diversos parâmetros definidos pelo compositor, como mudanças no tempo e de articulações, são notadas de maneira textual e por isso podem ser interpretadas de maneira subjetiva pelo intérprete. Assim, entre essas variações presentes na construção de uma performance encontram-se os atributos que estão além dos notados pelo compositor, como, por exemplo, as variações timbrísticas.

Inicialmente pensamos no timbre como um con-

junto de macro características do som capazes de diferenciar uma nota tocada com mesma altura, duração e intensidade em instrumentos de duas famílias distintas. No entanto, o timbre também permite identificar dois instrumentos de uma mesma família, ou identificar dois instrumentistas tocando instrumentos idênticos, ou ainda identificar duas performances do mesmo instrumentista com o mesmo instrumento em dois momentos distintos de sua carreira.

A complexidade que envolve a identificação do timbre se estende também a tentativa de descrever um som. Nem sempre é possível comunicar as características de um determinado som por não existir um acordo quanto aos adjetivos utilizados para descrever o timbre. É comum utilizar determinados adjetivos, neste contexto denominados correlatos semânticos, para definir o som e, muitas vezes, tais correlatos remetem a outros sentidos que não a audição como pesado / leve, claro / escuro, doce, ardido, metálico entre outros.

Tentando esclarecer como a variação timbrística pode alterar a expressividade da performance instrumental, esta pesquisa propõe investigar os correlatos semânticos utilizados para descrever o som e associar tais correlatos a descritores de áudio e recuperação de informação musical (MIR) de maneira a permitir uma a verificação formal de características timbrísticas. Uma vez que esta variação pode ser verificada formalmente, esta pesquisa propõe ainda o uso de aprendizado de máquina para classificar e agrupar estas variações a partir de gravações de performances musicais realizadas por profissionais. Esperamos com isso alcançar um maior entendimento das características timbrísticas do som para a performance musical, criar um modelo para sua análise, criar um conjunto de dados relacionados ao timbre de instrumentos e encontrar correlatos semânticos na língua portuguesa para as características timbrísticas do som [18].

2.7. Estudo do legato na clarineta através de parâmetros acústicos e cinemáticos

A clarineta, como os demais instrumentos de sopro, funciona por meio de feedback acústico permitindo ao músico controlar o fluxo e a manutenção da coluna de ar no interior do tubo; a variação da ressonância do tubo é ajustada e alterada de acordo com a abertura e fechamento dos orifícios do instrumento pela ação dos dedos do clarinetista [19]. Quando os movimentos dos dedos são realizados enquanto o músico sopra continuamente, há mudança de frequência e o efeito musical na transição de notas chamado legato. Estudos anteriores [20] mostraram que as variações na realização do legato dependem da qualidade do instrumento, das características acústicas do ambiente e da habilidade do músico. Partindo do pressuposto de que essa habilidade vem da forma como o músico manipula o trato vocal através da relação entre a posição da língua e a velocidade do ar, bem como a forma como movimenta os dedos, seguindo as afirmações de Stein [21] e Silva [22], pretendeu-se analisar a qualidade do legato na performance de clarinetistas através de parâmetros acústicos e cinemáticos.

Oito músicos profissionais gravaram excertos do

repertório orquestral onde a clarineta tem papel de destaque. Cada excerto foi executado repetidas vezes até que o(a) instrumentista decidisse que haviam sido gravadas quatro execuções consideradas satisfatórias; sem o uso do metrônomo e simulando uma situação real de concerto. Foi usado um microfone condensador cardióide M-audio Solaris disposto a cerca de um metro do(a) instrumentista e usada uma taxa de amostragem para a análise de 44.100 Hz. Foi usado o sistema óptico ativo de captura de movimentos em tempo real Optotrak com marcadores dispostos nas unhas dos músicos. As transições em legato selecionadas foram analisadas através do fluxo espectral, da energia RMS do sinal e do Índice de Legato do EXPAN [23]. Paralelamente, foi aplicado um teste subjetivo do tipo ABX com quarenta clarinetistas dos quais 14 são estudantes e 26 são profissionais.

Os resultados preliminares apontam que há um limite aceitável de assincronia entre os dedos e que essa assincronia pode definir a qualidade do legato nas transições. O teste ABX apontou que os estudantes concordaram percentualmente mais entre si do que os profissionais no ranqueamento das transições no que diz respeito à qualidade do legato e que, das doze combinações possíveis das transições analisadas no referido teste, os estudantes discordam dos profissionais em apenas duas combinações.

As conclusões indicam que a técnica e a preocupação com os movimentos dos dedos são fatores que não devem ser negligenciados no ensino do instrumento e que a percepção e avaliação das transições em legato, apesar de serem elevadas com a experiência, não são habilidades exclusivas de clarinetistas profissionais

2.8. Análise da consistência de legatos, ritmos e articulações de clarinetistas na Primeira Rapsódia de Debussy

Músicos profissionais e estudantes dedicam horas de estudos para adquirirem domínio sobre obras do repertório de seu respectivo instrumento. Estudam intensamente precisão rítmica, qualidade de articulações entre outros vários aspectos, necessários a uma boa interpretação. Com o tempo, vão sendo definidos padrões pessoais de qualidade desses aspectos, dada a singularidade de cada indivíduo, que se expandem para o repertório geral.

A Primeira Rapsódia para clarineta e piano de Claude Debussy [24] é uma das obras mais relevantes do repertório de clarineta, pela sua importância histórica e frequência em que é executada em recitais, concertos e processos de seleção [25, 26]. Nela, Debussy explora características do timbre do instrumento, sua extensa gama de dinâmica e de alturas, suas possibilidades de ligadura em grandes saltos intervalares e sua grande variedade de articulações.

A proposta desse trabalho é analisar a consistência interpretativa de clarinetistas profissionais e estudantes, atuantes na cidade de Belo Horizonte, na execução de excertos selecionados desta obra, que contêm dificuldades técnicas específicas e acompanhamento de piano. A um

pianista experiente será solicitado que toque os acompanhamentos de cada trecho selecionado e a cada clarinetista recrutado será solicitado que execute repetidamente sozinho cada trechos. Todas as execuções serão registradas em áudio. Em seguida, será solicitado aos clarinetistas que executem cada excerto repetidamente, ouvindo o acompanhamento da gravação prévia do piano, através de um fone auricular. A detecção de onsets e offsets de cada nota para a segmentação das notas será realizada com a utilização do sistema IRACEMA, desenvolvido no CEGeME para a análise empírica da performance musical [27].

O perfil das proporções rítmicas e da qualidade de articulação das notas e legato serão comparados entre as execuções com e sem acompanhamento de piano, visando verificar a influência do acompanhamento do piano na manipulação da proporção rítmica e da articulação, assim como da consistência interpretativa do clarinetista inferida a partir das repetições de execuções solo.

2.9. Aplicações sonológicas ao aprendizado e desenvolvimento musical

Sistemas computacionais mediam uma parte importante de nossas atividades diárias e se tornaram um componente essencial em praticamente todos os cenários que envolvem produção ou consumo de música. Os processos de ensino e aprendizagem musicais também sofreram tal influência, com o desenvolvimento de uma vasta gama de ferramentas de hardware, software, jogos e comunidades online a eles dedicadas. Quando tratamos de questões tradicionalmente consideradas mais subjetivas, como a expressividade musical, e em especial no ensino da música de concerto de tradição europeia, observamos certa resistência a tais abordagens.

Neste projeto investigamos uma série de oportunidades oferecidas pelos recentes avanços em tecnologias de gravação, processamento e reprodução de sinais, em conjunto com o crescente alcance das comunidades e plataformas online, no ensino e difusão da expertise musical. Foram realizados estudos com turmas de graduação de um curso de bacharelado em clarineta. Disciplinas foram ofertadas, com foco no desenvolvimento de habilidades ligadas à expressividade e performance de conjunto, através do estudo das técnicas necessárias para a execução de uma seleção de excertos de peças dos períodos clássico e romântico. Registramos em áudio todos os encontros semanais entre alunos e mestre, com duração de duas a três horas. Por meio de técnicas de processamento digital de sinais e extração de informação musical, foi possível obter uma série de descritores acústicos de tais gravações.

Pretendemos demonstrar como certos aspectos do desenvolvimento das técnicas necessárias para a realização de manipulações sonoras associadas à expressividade musical podem ser modelados computacionalmente, assim como elementos que fundamentam a relação mestre-aprendiz no contexto do ensino da música de concerto. Visamos também, a aplicação de métodos desenvolvidos neste projeto no desenvolvimento de ferramentas que possam enriquecer e facilitar o acesso ao conhecimento e ex-

pertise musical, especialmente em locais e comunidades onde o mesmo é escasso ou inexistente. Toda a tecnologia envolvida depende apenas de um computador pessoal convencional rodando software livre, disponibilizado gratuitamente nos sites de seus desenvolvedores.

Não temos como intenção argumentar que tais tecnologias possam vir a substituir um profissional do ensino da música. Acreditamos no entanto que, com tais técnicas à disposição, professores experientes poderiam não apenas aprimorar sua relação e comunicação com seus estudantes e o controle sobre seus avanços, como também incorporar outras referências de ensino e prática musical em seu trabalho.

2.10. Iracema: uma biblioteca Python para auxiliar os estudos empíricos da performance musical

Os primeiros estudos empíricos em performance musical datam do final do século XIX, quando foram criados os primeiros dispositivos mecânicos capazes de gravar as ações de pianistas no instrumento (o pressionar das teclas). Desde então, várias tecnologias que abrem novas possibilidades de coleta de dados de performances musicais foram inventadas ou aprimoradas, incluindo técnicas de extração de informação a partir do sinal de áudio. Tais técnicas, que se aprimoraram em ritmo acentuado ao longo das últimas décadas, impulsionadas pelo rápido desenvolvimento das mais diversas áreas correlatas à computação, são o foco do presente trabalho.

Propomos aqui uma nova biblioteca de software chamada Iracema[27], que contém técnicas para a extração de padrões de duração, energia, e conteúdo espectral, a partir de gravações de áudio monofônicas. Escolhemos a clarineta como o instrumento a ser utilizado nos experimentos de referência e modelos propostos, mas a maior parte das técnicas aqui apresentadas pode ser aplicada em outros instrumentos monofônicos.

Um dos passos mais importante no estudo de performances musicais é a detecção dos instantes de *onset* (início) das notas musicais, já que a nossa percepção rítmica (temporal) está fortemente associada a tais instantes. A este assunto dedicamos atenção especial, e propomos uma interface *web* para a anotação manual precisa dos instantes de onset, além de realizar um experimento para avaliar o erro típico de anotação neste tipo de tarefa, para gravações de clarineta.

Também propomos uma base de dados anotada contendo aproximadamente 23 minutos de áudio tocados na clarineta, contendo um total de 3551 onsets. Utilizando esta base de dados, treinamos uma rede neuronal convolucional para obter um modelo especializado para detecção automática de onsets em gravações de clarineta, e comparamos os seus resultados com os de outros modelos. Por fim, exemplificamos e demonstramos o uso da biblioteca proposta por meio de um estudo de caso, envolvendo a análise de gravações de um excerto de uma peça, tocada por vários clarinetas.

2.11. Individualidade, diversidade e consistência musical, e coesão na performance da música de câmara

Com foco no estudo empírico da performance musical, o projeto busca compreender os processos envolvidos na produção e na percepção da performance, a partir da identificação da intenção expressiva do intérprete encapsulada na sua manipulação do material acústico e dos movimentos corporais. Buscando consolidar nossos modelos analíticos da performance musical, a presente proposta avança sobre outros aspectos da prática musical, a performance de música de câmara, visando uma melhor compreensão de como músicos buscam coesão musical na interpretação de uma peça de música de câmara, manifestada pela sincronia entre as notas e pela homogeneidade de intensidade, qualidade sonora e de articulação. Nesta etapa pretendemos abordar questões sobre a performance musical em conjunto, que resultados anteriores fizeram emergir: (1) a evidência de graus distintos de consistência interpretativa entre músicos participantes, na realização de diferentes performances de uma mesma música; (2) a evidência de padrões distintos de sincronização e de acoplamento interpretativo entre os músicos tocando, que pode indicar a existência de habilidades específicas de cada músico para tocar em conjunto, seja como líder ou acompanhador.

O presente estudo pretende verificar correlações e efeitos da individualidade e da consistência interpretativa de um músico na coesão da performance de um conjunto musical e com que eficiência ele venha alcançar esta coesão. Poderia esta consistência contribuir ou influenciar na coesão de uma performance de música de câmara? A eficiência no alcance desta coesão (número de ensaios necessários) poderia também ser influenciada por estes fatores? Entender como músicos acoplam a interpretação da partitura, quando tocam em conjunto pode nos conduzir a uma melhor compreensão dos processos envolvidos na produção e percepção da expressividade de uma performance musical.

Estudos anteriores sobre a prática musical em conjunto utilizaram material musical de excertos sinfônicos de duos em uníssono, visando oferecer uma situação experimental que exigia a sincronização de cada nota com igualdade de condições para ambos os participantes. O presente estudo propõe experimentos que envolverão performance de excertos de obras de música de câmara, que buscarão trazer para o ambiente de experimentação situações mais complexas, com a introdução de elementos da estrutura musical, tais como estruturas harmônicas, rítmicas e de densidade sonora, que induzam em maior grau manifestações de expressividade.

3. Considerações Finais e Perspectivas Futuras

Este artigo expõe a variedade das questões tratadas nos estudos conduzidos no CEGeME, a qual revela o perfil essencialmente interdisciplinar da equipe de pesquisadores do Grupo. As diferentes áreas do conhecimento trazidas

por cada membro favorecem a emergência de um espaço de experiências coletivas que promove a aproximação e a articulação entre os diferentes campos disciplinares privilegiando a migração de conceitos e metodologias entre esses múltiplos campos. A inserção das atividades desenvolvidas no CEGeME no ensino, dado seu vínculo com a Pós-Graduação, se eleva a um patamar de formação diferenciada que capacita profissionais a frequentar áreas do conhecimento além de suas especialidades.

Resultados da pesquisa do CEGeME mostram perspectivas de oferecer uma percepção estendida de características e qualidades relacionadas com a expressividade musical, que poderia ser direcionada à prática musical cotidiana de instrumentistas e cantores, seja como recurso pedagógico em aulas de instrumento e canto, ou para contribuir para a própria busca pela excelência de sua performance. Tendo em vista tal perspectiva, prevemos o desenvolvimento de uma plataforma apropriada que possa oferecer a músicos profissionais, professores e alunos instrumentistas, uma interface interativa, que demande do usuário baixo investimento de treinamento, visando fornecer uma ferramenta mais simples e confortável para a pesquisa empírica da performance musical, que facilite a utilização por um maior número de usuários e para um maior volume de dados. Acreditamos que uma ferramenta como esta, devidamente testada e validada em situação real de prática e pedagogia musical, deve apresentar grande potencial de aplicação em diversos setores da prática e do ensino de música. Os resultados deste projeto e eventuais recursos por ele produzidos poderão também despertar interesse na área de pesquisa em outras áreas da música, tais como composição interativa, educação musical e musicoterapia, além de outras artes como teatro, dança e educação artística.

Referências

- [1] Hugo B de Paula, Hani C Yehia, JA Vasconcelos, and Mauricio A Loureiro. Representing timbre dynamics of a musical instrument: Comparison between ga and pca. In *Proceedings of the 2004 14th IEEE Signal Processing Society Workshop Machine Learning for Signal Processing, 2004.*, pages 381–390. IEEE, 2004.
- [2] Loureiro Mauricio Alves Loureiro, Hugo B de 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 de PAULA. Timbre de um instrumento musical: caracterização e representação. *Per Musi*, 14:57–81, 2006.
- [4] Maurício Loureiro, Tairone Magalhaes, Davi Mota, Aluizio Oliveira, and Thiago Campolina. A retrospective of the research on musical expression conducted at cegeme. In Flávio Schiavoni, Tiago Tavares, Rogério Constante, and Régis Rossi, editors, *Proceedings of the 17th Brazilian Symposium on Computer Music*, pages 165–172, São João del-Rei - MG - Brazil, September 2019. Sociedade Brasileira de Computação.
- [5] Renato Rodrigues Lisboa and Maurício Alves Loureiro. O trombone baixo moderno: uma revisão da literatura. In Fausto Borém and Luciana Monteiro de Castro, editors, *di-*

- torial dos Diálogos Musicais na Pós-Graduação: Práticas de Performance N.4. In: *Diálogos Musicais na Pós-Graduação: Práticas de Performance N.4.*, pages 274–288. Selo Minas de Som, Belo Horizonte, MG, Brazil, 2019.
- [6] Renato Lisboa, Gustavo Oliveira, Thiago Campolina, and Maurício Alves Loureiro. The effect of the use of the second valve on the duration of legato note transition on bass trombone. In *Proceedings of the 14th International Conference of Students of Systematic Musicology (SysMus21)*, Aarhus, Denmark, November 2021.
- [7] Gustavo Machado Oliveira et al. *Análise do pulso labial no trompete por filtragem inversa*. PhD thesis, Escola de Música, 2020.
- [8] Maurício Alves Loureiro Gustavo Machado Oliveira and Davi Alves Mota. Análise do fluxo labial no trompete através da filtragem inversa adaptativa interativa. In *Anais do XXX Congresso da Associação Nacional de Pesquisa e Pós-Graduação em Música*, Campina Grande - PB - Brazil, 2020.
- [9] Gustavo Machado Oliveira, Davi Alves Mota, and Maurício Alves Loureiro. Analysis of lip flow in trumpet playing using iterative adaptive inverse filtering. In D Dayser, editor, *Proceedings of the 13th International Conference of Students of Systematic Musicology (SysMus20)*, pages 15–17, York, United Kingdom, September 2020.
- [10] Tiago Lima Bicalho Cruz and Mauricio Alves Loureiro. Analysis of the glottal and supraglottal configuration and acoustic data in countertenor singing. *Per Musi*, 1(40):1–19, Jun. 2021.
- [11] L. Fuks. Acústica e experimentação nas palhetas duplas. In *Coletânea de Trabalhos Apresentados no I Encontro Internacional da Associação Brasileira de Palhetas Duplas e II Encontro Nordestino de Palhetas Duplas*, pages 34–41. UFPB, 2017.
- [12] Ravi Shankar Viana Domingues. *Análise de parâmetros acústicos e psicoacústicos da sonoridade do oboé associados aos diferentes estilos de raspados de palheta*. PhD thesis, Escola de Música, 2018.
- [13] David Ledet. Oboe reed styles. *Theory and Practice (Bloomington 1981)*, 1981.
- [14] 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.
- [15] Campolina T. de A. M. *Caracterização de Ataques na Clarineta via Modelagem e Decomposição Espectral*. PhD Thesis, 2019, Universidade Federal de Minas Gerais - Escola de Música, 2019. https://repositorio.ufmg.br/bitstream/1843/35785/1/Tese_ThiagoCampolina.pdf.
- [16] Campolina T. A. M. and M. A. Loureiro. Description of monophonic attacks in reverberant environments via spectral modeling. In *Proceedings of the 14th International Symposium on Computer Music Multidisciplinary Research. Perception, Representations, Image, Sound, Music*, pages 534–543, Marseille, France, 2019. https://hal.archives-ouvertes.fr/hal-02382500/file/Proceedings_CMMR2019.pdf#page=540.
- [17] Campolina T. A. M. and M. A. Loureiro. Musical note attacks description in reverberant environments. In Kronland-Martinet R., Ystad S., and Aramaki M., editors, *Perception, Representations, Image, Sound, Music. CMMR 2019. Lecture Notes in Computer Science*, volume 12631, pages 225–247. Springer, Cham, 2021. https://doi.org/10.1007/978-3-030-70210-6_16.
- [18] Flávio Luiz Schiavoni. A importância do gesto na análise timbrística da performance musical. In *Artes do Corpo I Corpo das Artes - 14o. Congresso Internacional de estética Brasil*, pages 42–42, Ouro Preto - MG, 2019.
- [19] Thomas D Rossing and Folden B Stumpf. The science of sound. *American Journal of Physics*, 50(10):955–955, 1982.
- [20] MA Loureiro, R Borges, T Campolina, T Magalhaes, D Mota, and HB DE PAULA. Extração de conteúdo musical em sinais de áudio para a análise de expressividade. *Encontro da sociedade brasileira de acústica*, 22:146–152, 2007.
- [21] Keith Stein. *The art of clarinet playing*. Alfred Music, 1958.
- [22] Alexandre Pereira da Silva. Captura do movimento dos dedos de clarinetistas: descrição dos processos metodológicos. *Anais do SIMPOM*, 1(4), 2016.
- [23] Mauricio Alves Loureiro, Davi Alves Mota, Thiago Campolina, Hani Camille Yehia, and Rafael Laboissière. Acolamento interpretativo e sincronização em duos de clarinetas. *OPUS*, 18(2):181–206, 2012.
- [24] Claude Debussy. *Première rhapsodie*, 1910.
- [25] Anna Briscoe and James R. Briscoe. Debussy for clarinet solo: The music and the conservatoire context. In *Clarinet-Fest® 2001*, New Orleans, Louisiana, 2001.
- [26] Santiago ABAD. La “première rhapsodie” de claud de-bussy. *Revista de investigación sobre la creación musical y artística de la Asociación de Compositores e Intérpretes Malagueños*, pages 40–56, 2016.
- [27] Tairone Nunes Magalhaes, Felipe Brandão Barros, and Mauricio Alves Loureiro. Iracema: a python library for audio content analysis. *Revista de Informática Teórica e Aplicada*, 27(4):127–138, 2020.

COUT-e em práticas itinerantes durante 2020 e 2021

Douglas Silva¹, Lucas Zampar¹, Felipe Rodrigues¹, Cláudio Gomes¹

¹Departamento de Ciências Exatas e Tecnologias - Universidade Federal do Amapá (UNIFAP)
Caixa Postal 261 – 68.906-970 – Macapá – AP – Brazil

{douglasdasilvasousa8, lucas.26.zampar, leepe.fr}@gmail.com, claudiorogério@unifap.br

Resumo. Este artigo apresenta ações em andamento e finalizadas pelo grupo itinerante de Computação mUsical e Tecnologias Emergentes (COUT-E) da Universidade Federal do Amapá (UNIFAP) durante o período de 2020 e 2021. Comenta-se sobre os projetos Handy Hear, CongAr, Acalma-te, classificação de ritmos populares da Amazônia e sincronização rítmica entre fontes distintas em tempo real.

Palavras-chave — aprendizado de máquina, mir, sistemas em tempo real

Abstract. This article presents ongoing and completed actions of the itinerant group of Emergent and Music Computer Technologies (COUT-e) from Federal University of Amapá (UNIFAP) during the period 2020 and 2021. Comments about the projects Handy Hear, CongAr, Acalma-te (Call down), and rhythmic synchronization in real time between different sources.

Keywords — machine learning, mir, real time systems

1 Introdução

No final de 2020, formou-se o grupo itinerante de Computação mUsical e Tecnologias Emergentes (COUT-e) para o desenvolvimento de ciência, tecnologia atuante, principalmente, nas áreas de computação musical, ciência de dados, processamento de sinais, redes de computadores e inteligência artificial. Atualmente, o COUT-e é composto por 4 integrantes sendo: um professor e 3 alunos de iniciação científica do curso de ciência da computação da UNIFAP.

2 Projeto finalizado

2.1 Handy Hear

O HandyHear [1] consiste em um protótipo de instrumento digital musical que simula ações de uma flauta (ou qualquer timbre) a partir de movimentações no ar sem toque físico ao protótipo. Ou seja, HandyHear capta a distância da mão, ou outro objeto, transformando-a em informação sonora e MIDI com a possibilidade de controle de amplitude.

As informações de sonoridade e MIDI são adquiridas no próprio HandyHear, mas pode-se enviar dados MIDI pela comunicação de rede local sem fio. O HandyHear está configurado para o tocante de duas oitavas. Utilizou-se um sistema embarcado do modelo *Raspiberry Pi 2* com um sensor de distância *HC-SR04* a partir da linguagem C.

3 Projetos em andamento

3.1 CongAr

Baseado no HandyHear, o CongAr é um protótipo de simulação de percussão de congas. Este projeto ainda em desenvolvimento,

apresenta ambiente visual e informação sonora e MIDI. O CongAr é um dispositivo distribuído no modelo de comunicação servidor-cliente. O cliente é um arduino do modelo ESP8266 e dois sensores de distância US-100. Cada sensor de distância simula uma conga, dessa forma, o CongAr simula as congas aguda e média com os golpes musicais aberto e fechado.

O servidor recebe os dados de distâncias e transforma em informação sonora, MIDI e permite a visualização da movimentação gestual de notas tocadas. A forma de comunicação entre os clientes CongAr e o servidor é por modo em série, comunicação em rede local sem fio ou internet.

3.2 Classificador automático de ritmos populares da Amazônia

O projeto de classificação automática de ritmos populares da Amazônia tem como objetivo fomentar pesquisas com ritmos musicais populares com bastante relevância, criatividade, constante mudanças e irreverência musical, porém pouco conhecidos no cenário musical nacional e internacional.

Para este trabalho inicial, criou-se a base de dados musical ¹a partir da plataforma de streaming youtube dos ritmos musicais de: andino, brega, carimbó, cumbia, marabaixo, merengue, pasillo, salsa, e vaqueirada envolvendo os países do Brasil (principalmente região norte), Guiana Francesa, Venezuela, Colômbia, Bolívia e Peru. Em média, selecionou-se 100 faixas para cada ritmo. Realizou-se testes na proporção de 70-30 com os modelos computacionais XGB, KNN, SVM e MLP alcançando a acurácia de 67.62%, 74.12%, 71.35% e 76.13%, respectivamente. Para trabalhos futuros, pretende-se adicionar ritmos e faixas para garantir robustez dos modelos.

3.3 Acalma-te

Outro trabalho, em fase inicial, é o *Acalma-te*, modelo de algoritmo de composição musical em tempo real com interatividade. O usuário define várias propriedades musicais como: tempo musical, timbre, tonalidade, amplitude, campo harmônico tonal e estrofes. O usuário define a quantidade de faixas melodias e harmonias, formas e sequência de estrofes. As intenções de composições são baseadas no campo harmônico tonal gregoriano.

Durante a composição, a harmonia pode ser independente ou ainda configurada com dependência da sequência de notas tocadas na faixa de melodia. O *Acalma-te* permite que, durante a composição, em alguns momentos mude-se involuntariamente a tonalidade, amplitude ou qualquer outra configuração musical. Neste momento, não criou-se o modelo rítmico.

Cada faixa é simulada como um instrumento MIDI, dessa forma, pode-se configurar em qualquer programa de gravação sonoro com qualquer efeito de timbre. Ao final, gera-se os arquivos MIDI de cada faixa.

¹ Base de dados: <https://bit.ly/3keCFuK>

4 Conclusões

É importante ressaltar a dificuldade na realização os projetos pois quase todos necessitaram de algum recurso material, sendo custeados por recurso individual. O COUT-e está em processo de formação de capital humano com ações para motivar e cativar atuais e novos alunos, promovendo palestras ou minicursos diversificados que promovam melhorias interpessoal e intelectual.

Referências

- [1] Damián Keller, Cláudio Gomes, and Luzilei Aliel. The handy metaphor: Bimanual, touchless interaction for the internet of musical things. *Journal of New Music Research*, 48(4):385–396, 2019.

Relatório de Pesquisa NESCoM 2021

Leandro Costalonga¹, Marcus Vinicius das Neves²

NESCoM – Núcleo Espírito-santense de Computação Musical

Universidade Federal do Espírito Santo

¹ Campus São Mateus

BR 101 Km 60 – 29932-540 São Mateus, ES

² Campus Goiabeiras

Av. Fernando Ferrari, 514 - Goiabeiras, Vitória - ES | CEP 29075-910

leandro.costalonga@ufes.br, creed.mvmn@gmail.com

Abstract

The NESCoM is a multidisciplinary research centre formed by musicians, engineers and computer scientists. This paper reports the ongoing projects and developments over the last two years, thus it is an update over the research report published in 2019. As a Brazilian research group, with solid international collaboration, we have opted to intercalate the language used to write the report, therefore the 2021 version is written in Portuguese. The main projects developed in these two years' timeframe are related to interaction design based on (bio)musicality, robotic music performance, and ubiquitous music. In addition, a strong artistic production is also described. If you are interested to get to know more about the projects, do not hesitate to contact us.

1. Um Breve Histórico

Em 2010, o Comitê de Computação Musical (CECM) da Sociedade Brasileira de Computação (SBC) aprovou Vitória-ES como cidade-sede do XIII Simpósio Brasileiro de Computação Musical (SBCM). Naquela época, havia poucas iniciativas de pesquisa na área acontecendo nas universidades locais. Movido pela necessidade de organizar tal evento, foi formado um comitê interinstitucional local. A relação entre as instituições anfitriãs foi se fortalecendo ao longo do processo de organização do evento, a tal ponto que alguns pesquisadores decidiram colaborar mais amplamente logo após o evento e, assim, o NESCoM foi fundado. Porém, somente em 2013 foi oficialmente registrado como grupo de pesquisa vinculado à Universidade Federal do Espírito Santo (UFES) e passou a figurar do diretório de grupos do CNPq.

O grupo é formado por músicos, artistas visuais, engenheiros eletricitas, além de engenheiros e cientistas da computação. Os integrantes estão principalmente distribuídos entre o Campus São Mateus-ES (Computação) e o Campus de Goiabeiras (Artes), este último localizado na sede administrativa da universidade,

em Vitória-ES. O fato de o grupo estar geograficamente disperso produziu diversos desafios ao longo dos anos mas também nos preparou para a colaboração no atual momento pandêmico vivido em 2020 e 2021.

O grupo vem atuando com disciplinas curriculares e optativas nos cursos de graduação em Ciência da Computação, Engenharia da Computação, Licenciatura em Música, Bacharelado em Música, Habilitação em Composição/Ênfase em Trilha Musical, e Artes Visuais. O grupo também está inserido no Programa de Pós-Graduação em Artes, na linha de pesquisa Interartes. Ademais, diversos projetos de pesquisa e extensão são desenvolvidos pelos integrantes, dentre os quais destaca-se projeto de extensão denominado Grupo de Experimentação Sonora (GEXS), que é o braço performático do NESCoM e que é apresentado na sequência. Na seção 3, é apresentado um compilado dos principais projetos de pesquisa realizado no último biênio. Finaliza-se o artigo com a projeções e perspectivas futuras do grupo.

2. Produção Artística (GEXS)

O Grupo de Experimentação Sonora (GEXS) completa 10 anos de atuação no ano de 2021. Foi registrado oficialmente como projeto de extensão na Universidade Federal do Espírito Santo (UFES) no ano de 2013 e vem registrando atividades em duas linhas de trabalho: [1] Estudo e criação de performances individuais e coletivas dentro do campo da Música Experimental e das Artes Sonoras, implementada desde sua criação; [2] Estudo da linguagem sonora para obras audiovisuais, sobretudo para cinema e séries; desde 2015. Do ano de 2019 até o momento o GEXS desenvolveu uma atividade performática coletiva, compilou coletâneas de peças musicais desenvolvidas e compostas pelos alunos integrantes do grupo e, com o surgimento da pandemia no ano de 2020, se voltou de maneira virtual aos estudos coletivos sobre técnicas de composição para música experimental e recebeu diversos artistas em um formato de conversa livre como atividade de aprendizagem e troca informativa sobre as poéticas de cada convidado. Assim,

em 2019 o grupo foi convidado pela Profa. Dra. Aline Dias, do Departamento de Artes Visuais da UFES, para compor trilha musical para uma compilação de vídeos dos alunos da sua disciplina chamada "Vídeo II". A compilação se tornou um média-metragem de 50 minutos de duração intitulado *Filme Sono*. O projeto geral fora intitulado ao redor do sono e a trilha musical composta teve duas finalidades: [a] funcionar como trilha musical para a obra audiovisual que, posteriormente, foi apresentada na exposição homônima ao projeto, organizada na GAP - Galeria de Arte e Pesquisa da UFES; [b] apresentação da trilha musical ao vivo junto à exibição do *Filme Sono* no Cine Sesc Glória, na qual participaram sete integrantes do GEXS em uma formação diversa e mista envolvendo uso de laptops, guitarra, instrumentos percussivos, de cordas friccionadas e instrumentos de sopro, todos com processamento em tempo real.

Ainda em 2019, o grupo desenvolveu duas atividades de composição que se transformaram em discos: [a] Coletânea 2018-2019, que reúne obras desenvolvidas por cada integrante do grupo neste biênio, totalizando 12 peças musicais de 9 integrantes; [b] estudo e composição de peças de poesia sonora a partir de textos literários escolhidos pelos participantes, totalizando 5 peças e o disco deve ser lançado ainda no ano de 2021. Todos os discos se encontram na plataforma do grupo no Bandcamp e estão disponíveis para download.

O ano de 2020 foi marcado pelo replanejamento das atividades no âmbito virtual devido à pandemia de COVID-19, mesmo assim o grupo publicou dois discos: [1] O Casamento de Clarice e Bataille, uma radionovela de cinco episódios derivada de filme média-metragem homônimo (2017) - ambos projetos assinados pelas artistas Aline Dias e Júlia Amaral - no qual o GEXS contribuiu com a produção, composição e performance ao vivo de uma das trilhas musicais; [b] a Coletânea 2020 - Pandemia, disco que reúne obras individuais produzidas pelos membros do GEXS durante este ano, totalizando 15 peças musicais de 13 integrantes.

Foi organizado também um seminário de livre que recebeu seis convidados, todos artistas e pesquisadores atuantes nos campos da produção musical, pós-produção de som para cinema, paisagem sonora, rádio arte e música experimental, são eles: Marcel Dadalto, Ana Monte, Thais Aragão, Janete El Haouli, Gimu e Nanati Francischini.

Para além dos discos, projetos de composição e seminários o grupo segue se encontrando semanalmente para estudos ligados às temáticas técnicas e conceituais de seu interesse e seus integrantes também desenvolvem em seus Trabalhos de Conclusão de Curso a partir dos estudos desenvolvidos ou derivados dos assuntos abordados no GEXS.

Neste ano de 2021 o grupo já conta com 12 estudantes de graduação majoritariamente advindos do curso de Bacharelado em Música e também possui alunos dos cursos de Licenciatura em Música, Artes Visuais e Bacharelado em Cinema e Audiovisual, 2 mestrands em Artes pela UFES, 3 egressos na condição de membros externos no projeto, somando ao todo 15 integrantes.

3. Projetos de Pesquisa e Desenvolvimento

O NESCoM tem um histórico de colaboração com diversos outros grupos de pesquisa nacionais e internacionais, em especial com os membros da rede de pesquisa multidisciplinar denominado G-UbiMus, na qual também é integrante. O NESCoM esteve sempre presente colaborando na organização do Workshop on Ubiquitous Music, tendo a primeira edição ocorrido 2010. Em 2011 e 2014, o NESCoM organizou e recebeu o evento em Vitória-ES. Em 2019, o evento também foi organizado pelo NESCoM em colaboração com o NAP (UFAC) e o PRISM (Marceille, França) na forma de um workshop do *International Symposium on Computer Music Multidisciplinary Research (CMMR)*. Em 2020, mais uma vez, o NESCoM foi co-chair do evento sediado pela UFSB (Porto Seguro, BA), mas realizado de forma online devido à crise sanitária.

A música ubíqua é uma área de pesquisa interdisciplinar que combina metodologias da música, ciência da computação, estudos em criatividade, ciências humanas e engenharias. Neste contexto, as pesquisas do NESCoM ligadas ao tema abrangem desde o estudo filosófico e teórico dos processos criativos no fazer musical até a implementação prática de soluções que suportem esses processos.

Um dos principais conceitos explorados em 2019 foi o da (bio)musicalidade, em especial aplicado nos projetos de interação de *Digital Musical Instruments (DMI)* e nos sistemas de música ubíqua[1]. Nesse sentido, mas em termos mais filosóficos, discutiu-se ainda o impacto da utilização massiva de recursos tecnológicos no fazer musical, principalmente nos mecanismos cognitivos que sustentam a capacidade humana inata de fazer música[2].

Um dos princípios bilíngues da música ubíqua é propiciar meios de realizar atividades musicais à indivíduos sem o devido treinamento formal em música, de tal sorte que o próprio conceito de música é mais ampliado e inclusivo. Assim sendo, não faria sentido simplesmente trocar a formação necessária para o domínio de um instrumento musical pela formação necessária para operar um sistema computacional musical. A interface e método de interação é, portanto, central ao conceito. Quanto mais intuitiva e pertinente aos processos biológicos inatos que permeiam o comportamento musical (musicalidade), menor se imagina que seja a curva de aprendizado, ou seja, mais natural seria a expressão da musicalidade do indivíduo ou do coletivo. Para verificar essa hipótese, o grupo vem trabalhando uma sintaxe de linguagem de programação musical para controle de robôs musicais que, ao mesmo tempo que substitui as habilidades motoras lentamente adquiridas por um instrumentista por movimentos robóticos de alta precisão, ainda permite uma programação hipoteticamente mais natural. Essa linguagem a parte final do projeto RoboMus que vem em constante desenvolvimento desde 2015 [3].

A produção de música tem sido feita predominantemente por meio de interações presenciais. Em caminho oposto, a pesquisa ubimus objetiva propor formas mais acessíveis de fazer música aliadas a uma busca por novas modalidades da prática artística, o que inclui o arcabouço teórico e aparatado técnico para que

atividades musicais possam ser feitas, por exemplo, assincronamente e remotamente. Nesse sentido, apropriamo-nos dos conceitos da Internet das Coisas (IoT) no contexto da música ubíqua para habilitar que atividades criativas possam usufruir da infraestrutura IoT, resultando em uma área de pesquisa denominada IoMusT – Internet of Musical Things [4]. Na prática, ioMusT pode ser descrita como o uso da eletrônica (sensores, atuadores, conectividade e etc.) para propósitos musicais. Uma aplicação englobando essas duas áreas de pesquisa é o desenvolvimento de instrumentos musicais robóticos para performances em conjunto, como se fossem parceiros ativos em ecossistemas musicais ubíquos [5].

O desenvolvimento de robôs musicais ainda permite explorar questões duas das vertentes de estudos atuais dentro do grupo: [1] acessibilidade; [2] prototipagem sustentável. A acessibilidade é a qualidade de um sistema, na qual a interface e a interação não excluem o público com necessidades especiais; o termo está relacionado ao acesso para todos, ou seja, é a possibilidade de qualquer pessoa, independentemente de suas capacidades físico-motoras e perceptivas, culturais e sociais, usufruir os benefícios de uma vida em sociedade portanto não pode ser característica alheia ao desenvolvimento de tecnologias voltadas a música ubíqua.

Os trabalhos com foco em acessibilidade no grupo se iniciaram ainda em 2018 com o desenvolvimento de uma interface computacional para controle musical que funcionava por meio do movimento dos olhos o usuário no controle da execução sonora[6]. O sistema, inclusivo também aos leigos no campo musical, fornecia indicações de escolhas de samples em uma paleta que auxiliava o usuário na performance musical. As indicações podiam ser configuradas de forma manual ou automatizada com auxílio do algoritmo de agrupamento K-means.

As pesquisas nessa estratégia de análise de característica visando seleção de sons com fins composicionais ou performáticos evoluíram no último biênio. Os avanços tecnológicos, principalmente o crescimento da Internet, permitiram o acesso a um enorme banco de dados de música digital, que contribui para uma nova forma de criação musical, inspirada no mashup e na estética musical concreta. Assim, as técnicas de recuperação de informações musicais (MIR) puderam ser aplicadas para encontrar samples que se combinassem para uso, tanto nessas interfaces assistivas, como no auxílio em possíveis performances musicais [7]. Esse tipo de uso é suportado pelas teorias que embasam os processos criativos adotado em música ubíqua onde a ordem das atividades e os tipos de recursos apontam para formas distintas de engajamento. Três tipos de atividades são inicialmente propostas: projeto, seleção e entrega (*deployment*). O trabalho descrito é um exemplo de aplicação da seleção e entrega de blocos musicais que podem ser utilizados em performances musicais ubíquas.

Como já mencionado, muito das pesquisas atuais do grupo circundam o desenvolvimento e uso de robôs em performances musicais – desde o projeto e construção dos robôs até o projeto de interação suportando as variadas práticas ubíquas e processos criativos. Atualmente, devido a dificuldade de acesso aos laboratórios de eletrônica em razão das medidas de isolamento social, o foco está na

virtualização dos robôs.

Nas últimas décadas, os robôs promoveram possibilidades únicas de performance e composição musical, permitindo novas interações com músicos e experiências memoráveis para o público. Os músicos robóticos podem ser construídos em muitas formas e ter diversas funcionalidades, tornando a musicalidade dos robôs um campo fértil de pesquisa. Porém, a construção de robôs físicos requer acesso a componentes elétricos e mecânicos, além de equipamentos de laboratório, o que pode inviabilizá-los financeiramente em países periféricos. Além disso, construir dispositivos experimentais físicos rapidamente levanta o problema de descartar peças quebradas ou desatualizadas. A preocupação com o desenvolvimento sustentável de novas tecnologias ligadas a música têm sido uma constante no grupo.

Equipamentos elétricos e eletrônicos (EEE) se tornaram uma parte essencial da vida cotidiana. Sua disponibilidade e uso generalizado permitiram que grande parte da população global se beneficiasse de padrões de vida mais elevados. No entanto, a maneira como produzimos, consumimos e descartamos o lixo eletrônico é insustentável. Em 2019, o mundo gerou 53,6 milhões de toneladas métricas (Mt) de lixo eletrônico, e apenas 17,4% desse montante foi oficialmente documentado como devidamente coletado e reciclado. Houve um aumento de 1,8 Mt de material reciclado desde 2014, mas a geração total de lixo eletrônico aumentou 9,2, indicando que as atividades de reciclagem não estão acompanhando o crescimento global do descarte de EEE. Isso não é diferente no contexto dos instrumentos musicais digitais (DMI), principalmente devido à sua cultura de prototipagem, onde o design e a construção de um (novo) instrumento costumam fazer parte da exclusividade da execução musical. Terminada a peça musical ou a temporada de performance, fica a pergunta: o que fazer com o equipamento?

Esses problemas estão relacionados ao “esverdeamento” da tecnologia, ou seja, aumentando a sustentabilidade ambiental, tornando todo o ciclo de vida do produto de tecnologias e produtos mais ecológicos, incluindo pesquisa, fabricação, uso e descarte. A Computação também estabeleceu sua parcela de contribuição para a preservação do meio ambiente sob o conceito de “Computação Verde”. Em termos simples, *Green Computing* envolve reduzir o impacto ambiental da tecnologia: isso significa usar menos energia, reduzir o desperdício e promover a sustentabilidade. Estamos agora preocupados com tais ideias de desenvolvimento sustentável em *Computer Music*, mais especificamente relacionadas ao desenvolvimento de DMIs. Assim, estamos interessados em como alguns conceitos fundamentais como *e-Waste*, *Ragpicking* e *Green Computing* podem ser aplicador para minimizar ou eliminar, sempre que possível, o impacto ambiental dos DMIs. Assim sendo, propomos e executamos ações relativas a três áreas que entende-se ter maior impacto nessa área: a) reciclagem de resíduos; b) reaproveitamento de materiais descartados; e c) uso de material sustentável e / ou fonte sustentável.

Um outro desafio quando se trabalha com robôs musicais advém da necessidade de sincronização. Robôs

têm atrasos mecânicos causados por componentes eletromecânicos. Esses atrasos precisam ser compensados para manter os robôs em sincronia. Assim, o grupo desenvolveu uma estratégia de aprender o padrão de atrasos mecânicos e compensá-los usando redes neurais em conjunto com a sincronização SNPT. Como prova de conceito, um o bongô robótico foi desenvolvido para testes. Os resultados experimentais para a técnica de compensação de atraso mecânico mostraram uma redução de 55,33% no atraso médio entre dois robôs[8].

O estudo de atrasos mecânicos (robôs) e biomecânicos (instrumentistas humanos) já foram objeto de amplo estudo nas pesquisas do grupo, principalmente na modelagem de performances musicais expressivas. Performances musicais geradas por computador - mesmo em um contexto ubimus - são frequentemente criticadas por serem incapazes de corresponder à expressividade encontrada em apresentações por humanos. Duas abordagens têm sido frequentemente adotadas para pesquisar a modelagem de desempenho musical expressivo em computadores. O primeiro enfoca o som; isto é, na modelagem de padrões de desvios entre uma performance humana gravada e a partitura musical. O segundo enfoca a modelagem dos processos cognitivos envolvidos em uma performance musical. Ambas as abordagens são válidas e podem se complementar. Nossa pesquisa focou no papel dos erros durante as performances musicais como fator de percepção da expressividade, ou seja, observou-se e modelou-se a manipulação física do instrumento pelo intérprete. Além disso, um estudo sobre os processos motores e biomecânicos da execução do violão foi proposto, em especial relacionados a velocidade, precisão, e força da mão esquerda de um violonista[9].

Atualmente, as pesquisas são desenvolvidas por 4 alunos de mestrado em Artes e 2 alunos de iniciação científica, que se somam ao corpo discente previamente apresentado no contexto do GEXS.

4. Conclusões e Perspectivas Futuras

Com já mencionado na última versão do relatório de pesquisa, publicado em 2019[10], o NESCoM continua sendo um grupo de pesquisa relativamente jovem, oficialmente fundado há pouco mais de 8 anos (2013). É um grupo multidisciplinar formado por pesquisadores de diferentes departamentos, incluindo ciência da computação, engenharia da computação, engenharia elétrica e, claro, música. Está geograficamente distribuído entre em dois campi da Universidade Federal do Espírito Santo, sendo que os cursos de Música, Artes e a Pós-Graduação em Artes estão localizados na sede em Vitória-ES (Campus Goiabeiras). Os cursos de Ciência da Computação e Engenharia da Computação no campus norte, em São Mateus, pouco mais de 220 km de distância.

O NESCoM também é membro do G-UbiMus, uma rede internacional de pesquisa multidisciplinar composta pela universidades UFRGS, UFAC, IFAC, UFSJ, UFES, Maynooth University (Irlanda), Linnaeus University (Suécia) e Queen Mary University of London (UK), entre vários outros colaboradores. O grupo continua empenhando esforços para melhorar a colaboração com outros grupos de pesquisa com interesses de pesquisa

semelhantes dentro da mesma universidade, como o [eMMA] Núcleo de Estudos em Música e Musicologia Auditável, inclusive absorvendo demandas de outras linguagens artísticas.

A atuação da Pós-Graduação em Artes trouxe novas demandas e renovou a ímpeto do grupo de aproximar as tecnologias das artes e vice-versa, assim como permitiu estabelecer novas parceiras, como com o Prof. Daniel Hora, que realiza pesquisas teóricas e práticas em arte e tecnologia, mídia, crítica e curadoria, com foco em estética, ativismo, cultura hacker, e sistemas generativos. Nesse sentido, um conjunto de novas disciplinas vem sendo ofertadas capacitando o corpo docentes a empreender esforços nesse sentido, tanto tecnicamente como artisticamente. A produção do grupo também passa por transformações, buscando maior diversificação e equilíbrio entre o desenvolvimento tecnológico e sua aplicação nas artes.

References

- [1] M. Costalonga, Leandro;Miletto, Evandro;Pimenta, “Musicality Centred Interaction Design to Ubimus: a First Discussion,” in Proceedings of the 14th International Symposium on Computer Music Multidisciplinary Research, 2019, pp. 640–651. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-02382500>
- [2] L. L. Costalonga, M. S. Pimenta, and M. M. Wanderley, “Can Ubimus Technologies affect our Musicality?,” *Per Musi*, no. 40, pp. 1–16, 2020, doi: 10.35699/2317-6377.2020.25941.
- [3] H. A. F. Camporez, T. S. R. Mota, E. M. v. Astorga, H. R. O. Rocha, and L. L. Costalonga, “RoboMus: Uma Plataforma para Performances Musicais Robóticas,” in *Aplicações em Musica Ublíqua*, 1st ed., D. Keller and M. H. de Lima, Eds. São Paulo, Brasil: ANPPOM, 2018, pp. 58–93.
- [4] D. Keller, L. Costalonga, M. Messina, D. Keller, L. Costalonga, and M. Messina, “Editorial: Ubiquitous Music Making in COVID-19 Times EDITORIAL Ubiquitous Music Making in COVID-19 Times,” 2020. Accessed: Jun. 23, 2021. [Online]. Available: <https://halshs.archives-ouvertes.fr/halshs-03035034>
- [5] H. Camporez, A. Garcia, J. Silva, L. Costalonga, and E. H. Rocha, “Internet das Coisas Musicais Aplicada a Instrumentos Musicais Robóticos (Internet of Musical Things Applied to Robotic Musical Instruments),” *International Journal of Digital Media and Interaction Special Issue on Ubiquitous Music*, vol. 3, no. 5, pp. 89–101, May 2020, Accessed: Jun. 24, 2021. [Online]. Available: <https://proa.ua.pt/index.php/jdmi/article/view/4262>
- [6] H. Araújo Fim Camporez, A. Frizzera Neto, L. Lesqueves Costalonga, and H. Roberto de Oliveira Rocha, “Interface Computacional para Controle Musical Utilizando os Movimentos dos Olhos,” 2018. Accessed:

Nov. 19, 2018. [Online]. Available: http://vortex.unespar.edu.br/camporez_et_al_v6_n2.pdf

[7] H. Camporez, Y. M. de Freitas, J. A. L. Silva, L. L. Costalonga, and H. R. de O. Rocha, “Features extraction and segmentation for an assistive musical interface,” Nov. 2020. doi: 10.5281/ZENODO.4248230.

[8] H. Camporez, J. Silva, L. Costalonga, and H. Rocha, “RoboMus: Robotic Musicians Synchronization,” Nov. 2020. doi: 10.5281/ZENODO.4247718.

[9] L. L. Costalonga, M. S. Pimenta, and E. R. Miranda, “Understanding biomechanical constraints for modelling expressive performance: A guitar case study,” *Journal of New Music Research*, vol. 48, no. 4, 2019, doi: 10.1080/09298215.2019.1643892.

[10] L. Costalonga, D. Coura, M. V. das Neves, F. Costa, and H. Rocha, “NESCoM Research Report (2019),” *Anais do Simpósio Brasileiro de Computação Musical (SBCM)*, pp. 151–154, Sep. 2019, doi: 10.5753/SBCM.2019.10437.

Workshops

musical tornou-se uma necessidade e linguagens de programação foram adaptadas porém, ainda não ideais para diversos cenários, permitindo o espaço para o surgimento de linguagens com diversos propósitos musicais. Dessa forma, C, C++, Python, Java e outras linguagens criaram bibliotecas com propósito musical, assim como o surgimento de linguagens computacionais e musicais como CSound, SuperCollider, Faust, Chuck, Processing, Over-tone, Pure Data, Max/SP, etc. [2, 3, 4, 5, 6, 7].

A linguagem de programação com propósito geral C pode ser utilizada para diversas ações tais como para a música. Na figura 3, apresenta-se um exemplo de codificação de geração sonora a partir da biblioteca *OpenAL* [8], na qual defini-se a frequência de execução ¹.

```

4 #include <stdio.h>
5 #include <stdlib.h>
6 #include <math.h>
7 #ifdef __APPLE__
8 #include <OpenAL/al.h>
9 #include <OpenAL/alc.h>
10 #elif _linux
11 #include <AL/al.h>
12 #include <AL/alc.h>
13 #include <unistd.h>
14 #endif
15
16 ALCdevice * openal_output_device;
17 ALCcontext * openal_output_context;
18 ALuint internal_buffer; // buffer sonoro
19 ALuint streaming_source[1]; // stream gerado
20
21 // Funcao padrao para erros
22 int al_check_error(const char * given_label) {
23     ALenum al_error;
24     al_error = alGetError();
25     if (AL_NO_ERROR != al_error) {
26         printf("ERROR - %s (%s)\n", alGetString(al_error), given_label);
27         return al_error;
28     }
29     return 0;
30 }
31
32 // inicializacao
33 void MM_init_all() {
34     const char * defname = alcGetString(NULL, ALC_DEFAULT_DEVICE_SPECIFIER);
35
36     openal_output_device = alcOpenDevice( defname );
37     openal_output_context = alcCreateContext( openal_output_device, NULL );
38     alcMakeContextCurrent( openal_output_context );
39     // setup buffer and source
40     alGenBuffers(1, & internal_buffer );
41     al_check_error("failed call to alGenBuffers");
42 }

```

Figura 3: Exemplo de geração sonora em C (p.1).

Percebe-se que em C, são necessários os seguintes passos para a geração de um som sintético: **inicialização** - definição do dispositivo sonoro, criação de um ambiente ou contexto, ativar o ambiente, criar áreas que receberão informações de áudio, os *buffers*, (*MM_init_all()*: linhas 38-50); **geração de fonte sonora** - definição temporal, taxa de amostragem, variável de *samples* conforme a dimensão definida, geração temporal da senoide (linha 80) com possíveis ajustes, atualização do *buffer* pelos *samples* de senoides, criação de transmissão e recepção do *buffer*, executar a transmissão (linha 96), análise de execução da transmissão do áudio (*MM_one_buffer(float)*: linhas 65-112); **término** - suspende a transmissão, retira dados da transmissão, deleta o ambiente de transmissão o contexto e dispositivo (*MM_exit_all()*: linhas 46-62). Há vários testes que estão comentados no código para a compreensão de outras possibilidades de ações com os *buffers*.

A linguagem de programação ChuckK [9] tem o propósito musical com funcionalidades de síntese, composição, análise e desempenho em tempo real. Uma das principais vantagens na utilização do ChuckK é a programação sincronizada ao tempo de precisão de controle musical além da adição

¹O arquivo completo pode ser acessado em: https://github.com/claudioorgerio/sintese_sonora

```

46 void MM_exit_all() {
47     ALenum error_code = 0;
48     // Stop the sources
49     alSourceStop(1, & streaming_source[0]); // streaming_source
50     int ii;
51     for (ii = 0; ii < 1; ++ii) {
52         alSource1(streaming_source[ii], AL_BUFFER, 0); }
53     // Clean-up
54     alDeleteSources(1, &streaming_source[0]);
55     alDeleteBuffers(16, &streaming_source[0]);
56     error_code = alGetError();
57     alcMakeContextCurrent(NULL);
58     error_code = alGetError();
59     alcDestroyContext(openal_output_context);
60     alcCloseDevice(openal_output_device);
61 }
62
63
64 // gerador de senoides
65 void MM_render_one_buffer( float freq ) {
66     float incr_freq = 0.06f;
67     int seconds = 4;
68     // unsigned sample_rate = 22050; // testes com diferentes rates
69     unsigned sample_rate = 44100;
70     double my_pi = 3.14159;
71     size_t buf_size = seconds * sample_rate;
72
73     // allocate PCM audio buffer vetor de samples
74     short * samples = malloc(sizeof(short) * buf_size);
75
76     printf("\nFreq senoide: %f\n", freq);
77     int i=0;
78     for (; i<buf_size; ++i) {
79         samples[i] = 32768 * sin( (2.f * my_pi * freq)/sample_rate * i );
80         freq += incr_freq;
81         if (100.0 > freq || freq > 2000.0)
82             incr_freq *= -1.0f; // toggle direction of freq increment
83         if (3000.0 > freq || freq > 5000.0)
84             incr_freq *= 1.0f; // toggle direction of freq increment
85     }
86 }

```

Figura 4: Exemplo de geração sonora em C (p.2).

```

87 // atualizando o internal_buffer com a senoide criada para o OpenAL */
88 alBufferData( internal_buffer, AL_FORMAT_MONO16, samples, buf_size, sample_rate);
89 al_check_error("populating alBufferData");
90
91 free(samples);
92
93
94 alGenSources(1, & streaming_source[0]); // cria um stream
95 alSource1(streaming_source[0], AL_BUFFER, internal_buffer); // adiciona buffer ao stream
96 alSourcePlay(streaming_source[0]); // play stream
97
98 // -----
99
100 ALenum current_playing_state;
101 alGetSource1(streaming_source[0], AL_SOURCE_STATE, & current_playing_state);
102 al_check_error("alGetSource1 AL_SOURCE_STATE");
103
104 while (AL_PLAYING == current_playing_state) {
105     printf("playing ... so sleep\n");
106     sleep(1); // should use a thread sleep NOT sleep() for a more responsive finish
107     alGetSource1(streaming_source[0], AL_SOURCE_STATE, & current_playing_state);
108     al_check_error("alGetSource1 AL_SOURCE_STATE");
109 }
110
111 printf("end of playing\n");
112
113 MM_exit_all();
114
115 }
116
117 int main() {
118     MM_init_all();
119     MM_render_one_buffer( 400.0 );
120 }
121

```

Figura 5: Exemplo de geração sonora em C (p.3).

de variáveis de manipulação temporal bastante úteis durante a programação. É uma linguagem multiplataforma integrada com outras linguagens de programação com grande flexibilidade, improvisação, permitindo o uso em diversos cenários artísticos.

A figura 6 apresenta um exemplo dos tipos de variáveis fundamentais úteis na codificação na linguagem ChuckK. Percebe-se que a principal mudança está na sintaxe orientada da esquerda para a direita. Além disso, há variáveis de controle temporal (variável *dur*, *time*) e do ambiente de contexto (variável *now*).

A figura 7 apresenta um exemplo a partir de funções de geradora de sinal impulso. Para esse exemplo, foi necessário passar o sinal impulso do filtro *BiQuad* para proteger e manter o ganho do sinal. Depois, avaliado pelo *Pan2*, que define a localização 2D da fonte sonora para a placa sonora (*Dac - Digital Analogic Conversor*). Após configurações do filtro *BiQuad* (linha

5-7), define-se a força de geração do impulso sempre constante, porém a frequência sonora e a localização da fonte sonora têm funções individuais aleatórias.

```
1 // variaveis
2 <<<"", "">>>;
3 <<<"", "">>>;
4 -1 => int i;
5 <<< i >>>;
6 1000 => i;
7 <<< i >>>;
8 0xA0 => i; // hexadecimal
9 <<< i >>>;
10 5.678 => float f; // float
11 <<< f >>>;
12 9.0::second => dur d;
13 <<< d >>>;
14 now => time t;
15 <<< t >>>;
16
17 <<< "Waiting 3 seconds..." >>>;
18 3::second => now; // stop de 3 segundos
19 60::ms => dur t_control;
20 <<< "Info:", now, t_control, now/t_control >>>;
21 0.1::second => now; // variavel de controle do tempo
22 <<< "Info:", now, t_control, now/t_control >>>;
23
24 0 => int j;
25 0 => int i;
26 do {
27   2 + => j;
28   1 + => i;
29   <<< i, j >>>;
30 } until ( i >= 5 );
31 <<< "Final", i, j >>>;
32 if ( j == 5 ) <<< "success" >>>;
```

Figura 6: Exemplo de variáveis em Chuck.

```
1 // sinal impulso para filtro para Posicao sonora
2
3 Impulse i => BiQuad f => Pan2 p => dac;
4
5 .99 => f.prad; // set filtro
6 1 => f.eqzs; // set equal gain zeros
7 .5 => f.gain; // set filter gain
8
9 while( true ) {
10   // forca do impulso
11   1.0 => i.next;
12
13   // frequencia aleatoria
14   Math.random2f( 250, 5000 ) => f.pfreq;
15   // direcao da fonte
16   Math.random2f( -1, 1 ) => p.pan;
17   // tempo de pausa
18   300::ms => now;
19 }
```

Figura 7: Exemplo de sinais sonoros em Chuck.

O Chuck apresenta diversas funções que facilitam o desenvolvimento e ações tais como gerar sons a partir de sintetizadores pré-configurados de STK (do inglês - *Synthesis ToolKit*). O STK é uma ferramenta de código de fonte aberto com vários timbres: flauta, sax, percussão, etc. Dessa forma, o Chuck tem variáveis na utilização do STK nativo, conforme o exemplo na figura 8.

A partir da linguagem Python, pode-se gerar ondas sonoras audíveis com biblioteca *IPython*. Conforme a figura 9, criou-se uma função que cria senoides e permite a soma de senoides baseada em posições na escala de notas musicais. Na linha 9, a variável *f0* tem a informação de frequência fundamental com base na equação de posição de notas musicais que será aplicada na função de senoide, linha 13. Na linha 26, executa-se a função descrita anteriormente, para em seguida gerar a visualização em 2D e ao final, o resultado para a fonte sonora, nas linhas 38 e 39.

Em resumo, cada linguagem de programação tem vantagens e desvantagens que deve-se avaliar qual o melhor paradigma de programação para utilizá-las em determinado cenário. É possível encontrar diversos exemplos, tutoriais nas referências

```
1 Flute flute => PoleZero f => JcRev r => dac;
2 .55 => r.gain;
3 .05 => r.mix;
4 .99 => f.blockZero;
5 // notas midis
6 [ 61, 63, 65, 60, 65, 66, 68 ] @=> int notes[];
7 while( true ) {
8   flute.clear( 1.0 ); // novas configuracoes aleatorias do instr
9   Math.random2f( 0, 1 ) => flute.jetDelay;
10  Math.random2f( 0, 1 ) => flute.jetReflection;
11  Math.random2f( 0, 1 ) => flute.endReflection;
12  Math.random2f( 0, 1 ) => flute.noiseGain;
13  Math.random2f( 0, 1 ) => flute.vibratoFreq;
14  Math.random2f( 0, 1 ) => flute.vibratoGain;
15  Math.random2f( 0, 1 ) => flute.pressure;
16  <<< "...", "" >>>;
17  <<< "jetDelay:", flute.jetDelay() >>>;
18  <<< "jetReflection:", flute.jetReflection() >>>;
19  <<< "endReflection:", flute.endReflection() >>>;
20  <<< "noiseGain:", flute.noiseGain() >>>;
21  <<< "vibratoFreq:", flute.vibratoFreq() >>>;
22  <<< "vibratoGain:", flute.vibratoGain() >>>;
23  <<< "breath pressure:", flute.pressure() >>>;
24  for( int i; i < notes.size(); i++ ) {
25    // duracao de cada nota aleatoriamente
26    Math.random2f( .75, 2 ) => float factor;
27    <<< "Tocar nota", notes[i] >>>;
28    play( notes[i], Math.random2f( .6, .9 ) ); // chama funcao play()
29    300::ms * factor => now; }
30 }
31 // funcao play
32 fun void play( float note, float velocity ) {
33   // midi para frequencia
34   Std.mtof( note ) => flute.freq;
35   velocity => flute.noteOn;
36 }
```

Figura 8: Exemplo de STK em Chuck.

```
1 import numpy as np
2
3 def timbre( steps, base, add, fs ):
4   pos = [0, 11, 23, 35, 47, 52]
5   ts = 1/fs # período de amostragem
6   duracao = 1 # em segundos
7   n1 = np.arange( fs*duracao ) # dominio do tempo discreto
8
9   f0 = base*np.power( 2, ((0+pos[steps])/12) ) # f0 de cada nota
10  amostra = np.sin( 2*np.pi*f0*n1*ts )
11  if add:
12    f0 = base*np.power( 2, ((0+pos[0])/12) ) # f0 de cada nota
13    amostra = np.sin( 2*np.pi*f0*n1*ts )
14    if steps >= 1:
15      for f in np.arange(steps)+1:
16        f1 = base*np.power( 2, ((0+pos[f])/12) )
17        amostra += (1/f)*np.sin( 2*np.pi*f1*n1*ts )
18      print( "Senoides somadas", f, pos[f], f0, f1 )
19  return amostra
20
21 import matplotlib.pyplot as plt
22 sr = 44100
23 harmonicos = 5
24 f0 = 220.0
25 somar_harmonicos = True
26 y = timbre( harmonicos, f0, somar_harmonicos, sr )
27 plt.figure( 1,figsize=(12,4) )
28 plt.subplot(121)
29 plt.plot(y)
30 plt.xlabel( 'Frame' )
31 plt.ylabel( 'Amplitude' )
32
33 plt.subplot(122)
34 plt.plot( y[0:800] )
35 plt.xlabel( 'Frame' )
36 plt.ylabel( 'Amplitude' )
37
38 from IPython.display import Audio
39 Audio( data = y, rate= sr )
```

Figura 9: Exemplo de sinais sonoros em Python.

de cada linguagem e suas respectivas bibliotecas.

3.1 Recuperação de Informação Musical – RIM

O tópico interdisciplinar de Recuperação de informação Musical – RIM (do inglês: *Music information Retrieval* – MIR) vem em continua evolução de ferramentas e modelos permitindo diversas aplicações e soluções científicas e mercadológicas. Atualmente, essas soluções são combinações com outros tópicos de suporte como: musicologia, psico-acústica, psicologia, processamento de sinais, aprendizado de máquina. Classificação de ritmos, sistemas de recomendações musicais, identificação de fontes musicais ou instrumentos, sincronização temporal, automática transcrição de músicas e geração musical são algumas das possíveis categorias em RIM.

Em RIM, basicamente extrai-se características do

áudio para utilização em um modelo definido. Dessa forma, em grande maioria, é necessário uma base de dados para entender as propriedades de cada áudio. A partir de uma fonte pode-se, basicamente, extrair características e identificar propriedades musicais e não-musicais, além de várias operações dependentes de cada solução. Há diversos tipos de extratores de características de uma fonte que tem funcionalidades similares tais como: MFCC – *Mel-Frequency Cepstral Coefficients*, RMS – *root-mean-square*, CENS – *chroma energy normalized*, CQT – *Constant-Q chroma-gram* etc. e demais extratores podem ser encontrados com mais detalhes em Lerch et. al. 2019 [10].

Há diversas bibliotecas ou ferramentas ágeis para auxiliar na extração de características de áudio. A biblioteca LibRosa [11] apresenta facilidade e ferramentas atualizadas. Conforme a figura 10, percebe-se que a diferença na utilização de cada extrator, está na própria chamada da função de extração de característica. Dessa forma, vale ao pesquisador aprofundar quais as características de fato são importantes para seu projeto.

```
1 import librosa
2
3 y, sr = librosa.load('file.wav', sr=None, mono=True)
4
5 print(y, sr)
6
7 four_tempo = librosa.feature.fourier_tempogram(y=y, sr=sr)
8 tempo = librosa.feature.tempogram(y=y, sr=sr)
9 chrom_stft = librosa.feature.chroma_stft(y=y, sr=sr, n_fft=2048, hop_length=512)
10 chrom_cens = librosa.feature.chroma_cens(y=y, sr=sr, hop_length=512)
11 chrom_cqt = librosa.feature.chroma_cqt(y=y, sr=sr, hop_length=512)
12 mel = librosa.feature.melspectrogram(y=y, sr=sr, n_mels=128)
13 mfcc = librosa.feature.mfcc(y=y, sr=sr, n_mfcc=20)
```

Figura 10: Extrair características de áudios em Python.

No entanto, quando tem-se a necessidade de realização de soluções para os tópicos de identificação de padrões seja para acordes, ritmos, emoções, timbres etc. tem-se em, uma abordagem, a necessidade de realização de extração de características de áudio de vários arquivos, geralmente confeccionados em uma base de dados. Logo em seguida, parte ao processo inteligível (regressões, lógicas, redes neurais, aprendizado de máquina etc.) que depende do cenário e das particularidades da base de dados. A conferência ISMIR, por exemplo, apresenta uma lista de várias bases de dados para diferentes atuações².

4 Conclusões

Neste minicurso, tentou-se cativar leigos e curiosos de diversas áreas para o tópico de computação musical. Para isso, utilizou-se a estratégia de inicializar fundamentos de teoria musical, características sonoras, ferramentas livres de edição e mixagem, programação para instrumentos musicais digitais de diferentes maneiras e por fim, apresentar características de recuperação de informação musical, um dos tópicos em grande movimentação no cenário internacional e nacional. Sugere-se consultas nas fontes no SBCM, Vortex Music Journal, ENCM, Revista Eletrônica de Iniciação Científica em Computação (REIC), International Society for Music Information Retrieval (ISMIR), International Country Music Journal (ICMC), Journal of New Research (JNMR), New Interfaces for Musical Expression (NIME), Ubiquitous Music Symposium (UBIMUS), Music Information Retrieval Evaluation eXchange (MIREX).

Referências

- [1] Forum Som ao vivo. Disponível em: <http://www.somaovivo.org>. Acesso em 10.09.2021.
- [2] Rodrigo Ramos de Araujo, José Mauro da Silva Sandy, Elder José Reoli Cirilo, and Flávio Luiz Schiavoni. Análise e classificação de linguagens de programação musical. *Revista Vórtex*, 6(2), 2018.
- [3] Clenio B Goncalves Junior and Murillo Rodrigo Petrucelli Homem. Um ambiente de programação multiparadigma voltado à composição musical automática. *Latin American Journal of Development*, 3(1):447–462, 2021.
- [4] Perry Cook. 2001: Principles for designing computer music controllers. In *A NIME Reader*, pages 1–13. Springer, 2017.
- [5] Felicia Nafeeza Persaud. *In Search of Computer Music Analysis: Music Information Retrieval, Optimization, and Machine Learning from 2000-2016*. PhD thesis, Université d'Ottawa/University of Ottawa, 2018.
- [6] Clemens Wöllner. *Body, sound and space in music and beyond: multimodal explorations*. Taylor & Francis, 2017.
- [7] Atau Tanaka and Miguel Ortiz. Gestural musical performance with physiological sensors, focusing on the electromyogram. 2017.
- [8] Garin Hiebert, K Charley, P Harrison, JM Jot, D Peacock, JM Trivi, and C Vogelsang. Openal programmer's guide. URL: <http://connect.creativelabs.com/openal/Documentation/Forms/AllItems.aspx>, 2007.
- [9] Ge Wang, Perry R Cook, and Spencer Salazar. Chuck: A strongly timed computer music language. *Computer Music Journal*, 39(4):10–29, 2015.
- [10] Alexander Lerch, Claire Arthur, Ashis Pati, and Siddharth Gururani. Music performance analysis: A survey. *arXiv preprint arXiv:1907.00178*, 2019.
- [11] 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*, volume 8, pages 18–25. Citeseer, 2015.

²ISMIR – Datasets: <https://ismir.net/resources/datasets/>

Oficina Música Visual Gerativa

Rute Moura¹, Jarbas Jacomé¹, Vinícius Garcia¹, Gustavo Carneiro¹, Horhanna Almeida¹, Mavi Pugliesi¹

¹Mustic* - Centro de Informática/Universidade Federal de Pernambuco

Av. Jornalista Aníbal Fernandes, s/n, Cidade Universitária (Campus Recife) – 50.740-560, Recife, PE

rutymaxsuelly@gmail.com, jjoj@cin.ufpe.br, vpg2@cin.ufpe.br, gtc@cin.ufpe.br, hao@cin.ufpe.br, mavipugliesi@gmail.com

Abstract. A computação expande as possibilidades na criação de obras artísticas de música visual, com a intencionalidade da associação algorítmica entre elementos do som e elementos gráficos para geração de visualizações. No entanto, há dificuldades interdisciplinares na exploração destes artefatos, pois as pessoas interessadas na temática nem sempre encontram direcionamentos para construí-los. Sendo assim, essa oficina surge pela necessidade de proporcionar a experimentação dessas potencialidades tecnológicas e tem por objetivo estabelecer um ambiente de criação colaborativa durante o SBCM 2021, propondo experimentação de códigos que integram o processamento de som e processamento e síntese de imagem. Utilizaremos um processo de Design colaborativo, considerando os participantes como protagonistas de seus experimentos pelo mapeamento de suas expectativas e interesses, auxiliados por tutorias e esperando obter como resultado esboços e contribuições experimentais de música visual gerativa.

1. Introdução

A multimedialidade cotidiana neste momento histórico, dada pela condição pós-mídia [1], parece reforçar em artistas de diferentes meios a vontade de experimentar as possibilidades da programação de computadores para criação. Uma vez que os elementos visuais e sonoros se tornam informação digital abre-se um universo específico de possibilidades de automação do cruzamento entre essas duas modalidades. Propomos esta oficina tendo em vista as necessidades de artistas que querem experimentar as possibilidades da computação para se expressarem através desses cruzamentos entre processamento digital integrado de som e imagem, mas não sabem por onde começar.

2. Música Visual

No artigo Foundations of Visual Music[2], Brian Evans define música visual como:

imagens visuais baseadas no tempo que estabelecem uma arquitetura temporal de maneira semelhante à música absoluta. É tipicamente não narrativa e não representacional (embora possa ser, em alguns casos). A Visual Music pode ser acompanhada de som, mas também pode ser silenciosa.[2]

A idéia de arte visual variada no tempo se aplica a atividades antigas, como teatros de sombra e

lanternistas viajantes, passando pelos órgãos de cor do século XVIII[3]. Na primeira metade do século XX, artistas como Mary Ellen Bute¹ apropriam-se de tecnologias industriais e do cinema para produção de música visual. Como demonstrados nas figuras 1 e 2 um recorte do filme *Tarantella* com suas experimentações como pioneira na animação e especialista em música visual.



Figure 1: Mary Ellen Bute



Figure 2: Quadros do filme *Tarantella* (1940) de Mary Ellen Bute

Com os grandes avanços tecnológicos em relação ao processamento gráfico de grandes volumes de informações, a música vem ganhando diversas formas de ser representada visualmente. Sendo assim, softwares de geração de visualizações dinâmicas surgem para auxiliar os artistas a complementar suas performances artísticas e ampliar as experiências do seu público. O *Iannix*² demonstrado na figura 3.

O *Iannix*, criado em homenagem a *Iannis Xenakis*, que em sua trajetória buscou realizar relações da matemática com a música, e com seus estudos contribuiu para o desenvolvimento da música computacional. O software se propõe a possibilitar experimentos na performance do artista com base no estímulo reativo fazendo-o explorar o espaço visual com traçados de segmentos de linha totalmente livres sobre um plano sonoro. Possui um processamento em tempo real de coleta dados sonoros com

*<https://mustic.cin.ufpe.br/>

¹<http://www.centerforvisualmusic.org/Bute.htm>

²Retirada de: <https://www.iannix.org/en/item/concerto-performance/>



Figure 3: Iannix- La Biennale Di Venezia

parâmetros do ambiente de programação, a exploração gráfica pela criatividade e interpretação de composições da sua estrutura musical é controlada e detectada por um dispositivo de captura de movimento que extrai eventos Open Sound Control (OSC) e gera a visualização sonora com ambientes 3D. [4]

3. Método da Oficina

Seguindo princípios pedagógicos de Paulo Freire [5], direcionaremos a oficina aos participantes enquanto protagonistas da experiência e da construção do próprio saber, incentivando-os a utilizar recursos computacionais para resolver problemas reais. Sendo assim, primeiramente dedicaremos um tempo para uma troca compartilhada através do discurso falado, contextualizando expectativas, vivências e interesses das pessoas para, junto com elas, propormos novos experimentos de música visual gerativa. Visando uma troca construtiva, conduziremos interações por discursos falados através de conferência online, aplicando métodos de design como *brainstorming* e criação de protótipos. A tabela 1 detalha uma proposta de cronograma de seis horas, sendo duas horas em cada um dos três dias de evento.

Table 1: Planejamento cronológico da oficina.

Primeiro dia de oficina	
00:00	Apresentação da oficina
00:10	Participantes falam sobre suas expectativas
00:40	Apresentação das propostas de experimentos
01:00	Discussões para escolha das propostas e divisão em grupos
Segundo dia de oficina	
02:00	Explicação dos códigos das propostas
02:30	Experimentação e tutorias
03:30	Apresentação de status com todo grupo reunido
Terceiro dia de oficina	
04:00	Experimentação e tutoria
05:00	Apresentação dos projetos

Utilizaremos um modelo de design experimental, como demonstra a figura 4, desenvolvido pelos proponentes dessa oficina como uma adaptação ao diagrama de afinidades descrito por Vianna [6], facilitando a integração de uma equipe multidisciplinar para abordar as temáticas e problemas reais.

Serão realizadas discussões para entender o perfil do público participante, desenvolvendo um processo



Figure 4: Modelo estrutural oficina

de aprendizado compartilhado e interativo através da apresentação de ferramentas de geração de música visual. A oficina será dividida em cinco momentos:

1. Introdução a visualização musical gerativa.
2. Discussão e adaptação adequada as realidades dos participantes e autores, em relação as propostas de experimentos a serem trabalhadas na oficina.
3. Proposta de experimentos e apresentação de ferramentas.
4. Definição e escolha da proposta prática, integração dos grupos segundo a escolha da proposta e execução e tutoria da proposta prática.
5. Discussão e apresentação dos projetos.

Visando facilitar a escolha da proposta prática, será realizado uma dinâmica de grupo, apresentando as propostas de experimentos utilizando um método popular para gerar ideias em grupo: o *brainstorming*. Esse método instrui membros do grupo a desenvolverem ideias anteriores, não criticarem nenhuma ideia, gerar um grande número de ideias e se sentirem livres para contribuir com ideias extravagantes [7]. Após a apresentação das propostas, os participantes utilizarão a plataforma de quadros colaborativos online Miro para registro de *brainstorming* e coleta colaborativa de *insights*.

Os autores também irão disponibilizar um material informativo na chamada da oficina para direcionamento aos participantes de como se dará seu desenvolvimento e com uma lista de ferramentas e instruções.

Para participação da oficina consideramos os seguintes pré-requisitos:

1. Interesse na exploração de ferramentas computacionais.
2. Acesso à internet e à plataforma *Discord* para participação dos encontros online.
3. Acesso a computador com entrada e saída de áudio, para processamento e acompanhamento do som dos experimentos de música visual gerativa.

4. Instalação e configuração do ambiente de *software* de acordo com instruções a serem divulgadas na chamada da oficina no *site* do evento SBCM 2021.

4. Propostas de experimentos

Tendo como objetivo incentivar o pensamento criativo e criar uma dinâmica colaborativa na criação dos experimentos, provocaremos os participantes com a pergunta "Se de uma forma simples e rápida pudéssemos dar uma forma ao som, como o representaríamos?"[8]. A partir do critério de imersão no contexto real, traremos quatro propostas de tutorias, para explorar o processo de criativo de geração automática de música através de ferramentas são elas: (1) Onda sonora com forma visual estável no tempo, (2) Visualizando som com Hydra, (3) Visualizando som com formas gráficas em p5.js e (4) Visualizações musicais com MIDI.

4.1. Onda sonora com forma visual estável no tempo

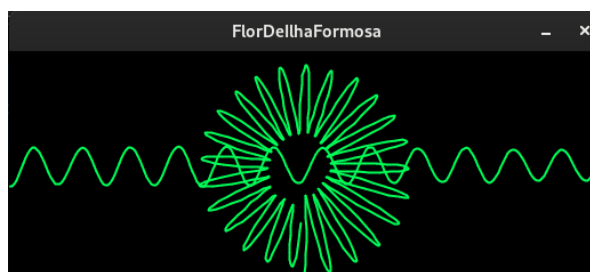


Figure 5: Onda sonora com eixo do tempo circular.

O desenho em tempo-real da onda sonora é uma das técnicas utilizadas para criação de efeitos de visualização de áudio. No entanto, ao plotarmos a onda sonora em tempo-real, a cada quadro as montanhas e vales da onda são desenhados em posições diferentes, gerando uma descontinuidade visual entre um quadro e outro da animação gerada. Jácome [9] apresenta um algoritmo simples para encontrar os "topos das montanhas" e plotar sempre a onda sonora com essas montanhas mais ou menos na mesma posição, ao longo do tempo. Partindo desta técnica podemos plotar a onda em diversos formatos, como por exemplo, considerando o eixo X (tempo) da onda como sendo uma circunferência. Fazendo isso temos como resultado uma flor (como na figura 5) cujas pétalas se mantém mais ou menos na mesma posição nos diferentes quadros. Uma implementação desse algoritmo pode ser acessada no GitHub³.

Pré-requisito específico: Instalação de IDE Processing⁴

4.2. Visualizando som com Hydra

Baseada nos sintetizadores analógicos de vídeo, utilizados principalmente de modo experimental por artistas desde os

³Disponível em: <https://github.com/jarbasjacome/ExperimentosDeProcessing/blob/main/MusicaVisual/FlorDeIlhaFormosa/FlorDeIlhaFormosa.pde>

⁴Disponível em: <https://processing.org/>

anos 70 [10], a linguagem de programação *Hydra*⁵ permite a criação e interligação de programas modulares que simulam os sintetizadores analógicos de forma que a criação de imagens torna-se facilmente explorável. O ambiente de programação Hydra é executado no próprio navegador e, portanto, não é necessária a instalação de nenhum programa para a exploração com essa ferramenta, representada na figura 6.

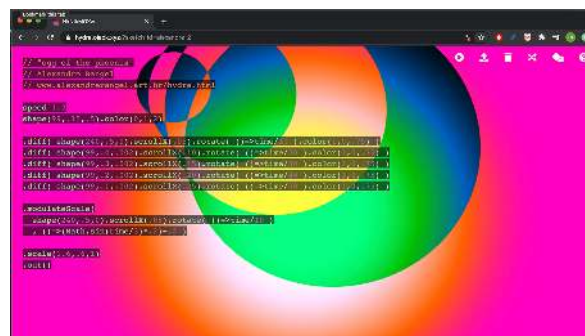


Figure 6: Hydra

4.3. Visualizando som com formas gráficas em p5.js

Com base em estudos preliminares, traremos experimentos já realizados pelos autores na biblioteca p5.js, demonstrando exemplares de visualizações sonoras, executando-as em tempo real para incentivar o potencial criativo dos participantes em relação utilização de formas gráficas como triângulos. Como por exemplo no experimento gráfico da figura 7, chamado de *Gotas Sonoras*⁶



Figure 7: Gotas Sonoras

Esta visualização sonora, possui um oscilador e realiza síntese sonora utilizando recursos computacionais, integrados a representações visuais e exploração de propriedades gráficas como cores, formas já mapeadas e experimentadas pelos tutores que terão como papel ser mediadores com compartilhamento de aprendizados práticos e demonstração de exemplos.

⁵Hydra: <https://hydra.ojack.xyz/>

⁶Disponível em: https://github.com/Rute123/Music_Visual/tree/master/Gotas_Sonoras

No desenvolvimento independente dos participantes, o acompanhamento das tutorias auxiliará na validação de ideias modificando e experimentando códigos abertos. Durante os experimentos práticos iremos demonstrar o ambiente de trabalho da biblioteca p5.js, propondo a utilização de seu editor web que pode ser acessado através do navegador onde realizaremos manipulação de funções gráficas e musicais baseadas na documentação e nas necessidades exploratórias de cada participante.

Pré-requisito específico: Acesso a internet para acesso a editor web p5.js⁷

4.4. Visualizações com MIDI

A visualização automatizada de dados musicais pode partir não apenas da análise do áudio, mas também de dados MIDI. Além de poder representar peças musicais sequencialmente, o protocolo MIDI permite a comunicação em tempo de execução com os programas a partir de interfaces (físicas) controladoras.

Sendo assim, ampliamos o espaço para experimentação. E propomos com esta tutoria fazer uma apresentação de exemplos de projetos de visualização musical a partir de dados MIDI, por exemplo o *Da Capo*⁸ demonstrado na figura 8 uma interface interativa que utiliza de geração dinâmica baseada em controladores MIDI coletando as informações em tempo real de acordo com a execução musical.

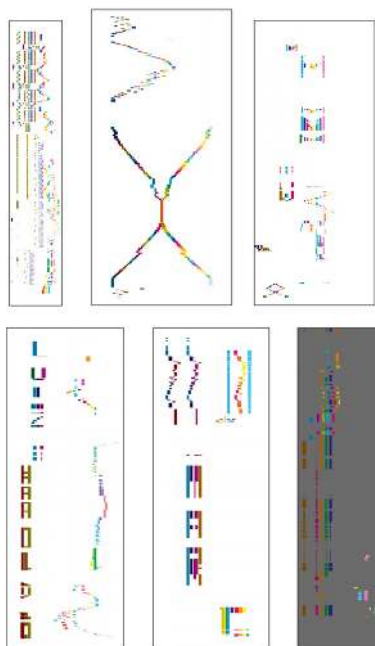


Figure 8: Da Capo

Além desses exemplos de visualizações dinâmicas e interativas existem diversos outros tipos, que podem ser imagens estáticas representativas de peças escritas em MIDI, vídeos ou até performances ao vivo. Na oficina também pretendemos abrir esta discussão

⁷Biblioteca p5.js: <https://p5js.org/>

⁸<http://www.life.dad.puc-rio.br/projetos/dacapo.html>

buscando quebrar as barreiras de comunicação entre as áreas e possibilitar experimentações práticas ao público participante a partir de projetos criados pelos autores.

Como por exemplo o *mandrit*⁹ demonstrado na figura 9, um sistema desenvolvido através de processamento de dados musicais com o Processing e com a biblioteca p5.js, buscando ampliar a comunicação visual da informação rítmica através da experimentação de processos criativos, extração de características do ritmo e a geração de visualizações estáticas de obras musicais.

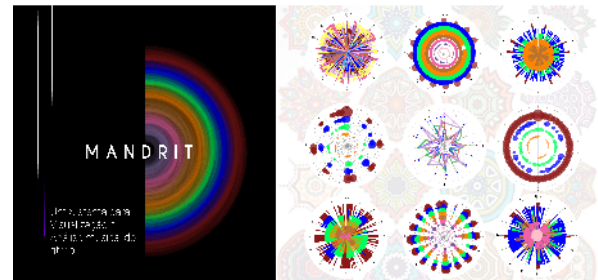


Figure 9: Mandrit

O *mandrit* extrai informações rítmicas de arquivos MIDI e transforma em assinaturas visuais, que auxiliam nos processos de análise e visualização musical do ritmo.

References

- [1] Peter Weibel. The Post-media Condition. *Arte ConTexto*, 6:11–15, 2005.
- [2] Brian Evans. Foundations of a visual music. *Computer Music Journal*, 29(4):11–24, 2005.
- [3] Henrique Roscoe. Tocando Imagens: Dispositivos e Técnicas da Visual Music. Master's thesis, Universidade Federal de Minas Gerais, 2019.
- [4] Julian Scordato. Composing with Iannix. In *Proceedings of the Fifth Conference on Computation, Communication, Aesthetics and X.*, volume V, page 389, Lisboa, Portugal, 2017.
- [5] Paulo Freire. *Pedagogia da autonomia: saberes necessários à prática educativa*. Paz e Terra, São Paulo (SP), 2006. OCLC: 817009516.
- [6] Maurício Vianna, Ysmar Vianna, Isabel K Adler, Brenda Lucena, and Beatriz Russo. *Design Thinking : Inovação em Negócios*. [recurso eletrônico]. MJV Press, Rio de Janeiro, 2012.
- [7] Vicky L. Putman and Paul B. Paulus. Brainstorming, Brainstorming Rules and Decision Making. *The Journal of Creative Behavior*, 43(1):29–40, March 2009.
- [8] Gonçalo Daniel Soares das Neves. *Visualização do Som*. Dissertação de Mestrado., Universidade de Coimbra, Departamento de Engenharia Informática da FCTUC, September 2017.
- [9] Jarbas Jácome. Música Visual, Algoritmo de Zarathustra e Flor de Ilha Formosa. In Lucas Bambozzi and Almar Rena, editors, *ENTRE ARTE E TECNOLÓGICAS: IDEIAS EM TRÂNSITO*, pages 46–53. Fluxos, 2015.
- [10] Thomas Dreher. *History of Computer Art*. IASLonline, 2014.

⁹Mandrit: <https://zenodo.org/record/4118575>

Towards a Latin American NIME Research Community

Juan Pablo Martínez Avila (Mexico)¹, João Tragtenberg (Brazil)², Filipe Calegario (Brazil)³, Juan Ramos (Argentina)^{4*}, Martín Matus Lerner (Argentina)⁵, Isabela Corintha (Brazil)⁶, Javier Jaimovich (Chile)⁷, Teodoro Dannemann (Chile)⁸, Patricia Cadavid (Colombia)⁹, Ximena Alarcón (Colombia)¹⁰, Adnan Marquez-Borbon (Mexico)¹¹, Hugo Solís (Mexico)¹², Miguel Ortiz (Mexico)¹³

¹Mixed Reality Lab, University of Nottingham (UK) - psxjpma@nott.ac.uk

²Programa de Doutorado em Design, Universidade Federal de Pernambuco (Brazil) - tragtenberg@gmail.com

³Centro de Informática, Universidade Federal de Pernambuco (Brazil) - fcac@cin.ufpe.br

⁴Universidad Nacional de Quilmes / CONICET (Argentina) - juan.ramos@unq.edu.ar

⁵Universidad Nacional de Quilmes (Argentina) - mmatus@unq.edu.ar

⁶Programa Doutoral Multimédia em Educação - Universidade de Aveiro (Portugal) - isabelaalmeida29@gmail.com

⁷Facultad de Artes, Universidad de Chile (Chile) - javier.jaimovich@uchile.cl

⁸Media and Arts Technology PhD Programme, Queen Mary University of London (UK) - t.dannemann@qmul.ac.uk

⁹Interface Cultures, Kunstuniversität Linz (Austria) - lpcadavid@gmail.com

¹⁰Independent Artist/Researcher - info@ximenaalarcon.net

¹¹Facultad de Artes, Universidad Autónoma de Baja California (Mexico) - adnan.marquez@uabc.edu.mx

¹²Departamento de Artes y Humanidades, Universidad Autónoma de México (Mexico) - h.solis@correo.ier.uam.mx

¹³Sonic Arts Research Centre, Queen's University Belfast (UK) - m.ortiz@qub.ac.uk

Abstract. *In this workshop, we propose the first steps towards consolidating a Latin American (LATAM) New Interfaces for Musical Expression (NIME) Research Community. We aim to start a discussion and generate ideas around short-term actionable initiatives that a LATAM NIME Research Community could explore in the near future. We will discuss various topics centered around three main areas: (1) The creation of a LATAM NIME network, (2) Exploring LATAM NIME perspectives, and (3) the Next Steps for LATAM NIME. After this activity, we will have a plenary discussion where we will share ideas for action and reflections. Lastly, we list a series of expected outcomes of the workshop, although we acknowledge that many others may emerge during the workshop. We look forward to taking the first steps towards a LATAM NIME Community and the future of this community within and beyond NIME, opening to other communities in LATAM, and stirring policy changes to support the development of our local communities.*

1. Workshop Description

Historically, Latin America (LATAM) has produced many musics that have resonated around the world. Equally, it has also significantly contributed and continues to contribute to the field of New Interfaces for Musical Expression (NIME) [1]. Yet, the NIME conference has only been hosted in LATAM once in 2019 at Porto Alegre, Brazil, where only 18 publications out of 88 were from LATAM authors and co-authors [2], and authors affiliated to South

American institutions¹ represent only 2.4% of the total amount of participants of the first 20 NIME Conference editions (from 2001 to 2020) [3]. In this workshop we aim to discuss ways of further visibilizing and fostering NIME research in, by and for LATAM, echoing recent efforts from the wider LATAM Human-Computer Interaction research community [4]. Moreover, we primarily expect to consolidate a LATAM NIME Research Community through a series of actionable initiatives, such as creating a LATAM NIME network, and exploring ways of making NIME research more accessible, affordable and resourceful in consideration of regional geopolitical and socioeconomic challenges, among other actions.

2. Workshop Activities

The workshop will last approximately 4 hours and it will begin with a general introduction, i.e., introducing the organisers and their research work, and the workshop goals and activities. After a short break we will then introduce three general discussion areas, i.e., (1) Creating a LATAM NIME Network, (2) Exploring LATAM NIME Perspectives, (3) Next Steps for LATAM NIME. These discussion areas will be further subdivided into a series of short-term and long-term actions, among other topics, e.g., sharing resources and knowledge, mentorship programs, etc. The workshop schedule will vary according to the number of attendees. If we have a large number of attendees we will split into breakout discussion groups to discuss all of the

*Supported by CONICET

¹The analysis considered only a division between North and South Americas, not considering the cultural/linguistic division of Latin America and didn't consider the authors nationality in the analysis

discussion areas in smaller groups. If we have a small number of attendees we will keep everyone in a single group and discuss all of the topics together. After the discussion activity, we will have a plenary discussion where we will share ideas and reflections. Then we will wrap up and conclude the workshop.

3. Discussion areas

The following discussion areas are meant for starting a discussion and generate ideas around short-term actionable initiatives that a LATAM NIME Research Community could explore in the near future. One of these initiatives involves creating communication channels, wikis, blogs and repositories to improve the communication and sharing of knowledge and resources amongst academics, artists, and makers, fostering a LATAM NIME network. We also intend to discuss how we can create a mentorship program to support new members of the community in the near future. Likewise, we intend to discuss long-term initiatives that can unfold over the years, such as the organisation and hosting of future NIME conferences in LATAM, and the creation of local NIME events, such as chapters and hubs, to further engage the LATAM community.

3.1. Creating a LATAM NIME Network

3.1.1. Building stronger ties

How can we build stronger ties between the different regions of LATAM making or wanting to make NIME research? We will discuss possible aims and goals of the network, as a group of researchers who may collaborate and self-organise together, as well as support each other and their communities, by exchanging knowledge, resources, and opportunities. This network may sustain itself through multiple channels such as a NIME subdomain², a mailing list, a forum, and social media platforms, such as Discord or Telegram, among others. These channels may facilitate the sharing of calls for funding and participation, open positions, and news or updates related to the community, as well as providing visibility to the works of its members, in areas such as DMI design [5, 6, 7, 8, 9, 10, 11], accessible DMIs [12, 13], bodily interactions [14, 15] and Somaesthetics [16, 17], through blog posts, interviews, and a directory (as in the WONOMUTE blog³). Another consideration to be discussed is how can we make this community more inclusive to people outside of NIME who may be interested in understanding, researching, learning, teaching, designing, developing and performing with DMIs and/or music technology in a broader sense. These people may be makers, artists, musicians, pedagogues, independent researchers, or members of the general public. Likewise, professionals in other areas of work, such as science, engineering, the arts, and mathematics (STEAM), and others, may be also interested in exploring, and knowing more about NIME and even contribute to it.

²<https://latam.nime.org/>

³<https://wonomute.no/>

3.1.2. Developing NIME Pedagogies for LATAM

We will discuss ways of sharing knowledge and educational content related to NIME, that may provide the basic skills for building digital musical instruments (DMIs), sourcing, synthesizing and designing sounds, and mappings for these instruments, and making music with them, among other relevant skills. Likewise, efforts can be made to translate existing knowledge bases into LATAM languages. The channels and format to share educational materials will also be discussed, as this could potentially be structured as online courses or workshops that can be imparted locally. The contents of these materials could also take into account local musical traditions and decolonized perspectives. Moreover, we shall discuss pedagogies with emergent musical technologies [18, 19, 20, 21] and how can these be informed by the rich history of education and pedagogy in LATAM.

3.1.3. Mentorship Programs for LATAM

How can we mentor students, early career researchers and new members to the NIME community or working in NIME-related subjects in LATAM? Here we may discuss how we can provide mentorship and supports to research institutions and organisations who may be interested in working with NIME but do not have the experience, frameworks or resources to do so.

3.2. Exploring LATAM NIME Perspectives

3.2.1. Decolonising NIME Research: a LATAM Perspective

How can we further visibilize NIME research happening in LATAM by local institutions and with local populations, considering the institutional, political, economic, ecosystemic, cultural and ethnomusicological factors of each region, as well as research by LATAM researchers abroad who contribute to NIME and/or to LATAM through their research and associated projects, either institutionally-supported or by their own initiative? How can we explore alternative narratives for the design of DMIs which are centred in the sociocultural reality of LATAM? These DMIs may be based on popular [22] or traditional instruments [23] and technologies [24, 25], musics and dances [26] of each region, or they may be designed by harnessing existing local approaches [27] or elements of the local cultures, offering new meanings, insights and design opportunities for NIME. They may also explore concepts and narratives of the LATAM worldview [28, 29].

3.2.2. Making NIME Research More Affordable and Resourceful

What are the ways in which we can make NIME research more affordable and resourceful in less privileged regions and institutions? This discussion may gravitate towards open source and low-cost software and hardware [22], recycling and repurposing electronic waste, tearing down unused or old electronic/digital instruments to learn from them, and enabling ways to exchange materials and knowledge [27]. This knowledge exchange might extend to other arts and disciplines, as well as to the general public and communities that can use and enjoy NIME.

3.3. Next Steps for LATAM NIME

3.3.1. Hosting More NIMEs in LATAM

NIME has been successfully hosted in Porto Alegre, Brazil, opening new possibilities for future NIMEs in LATAM. What are the considerations for hosting future NIMEs in LATAM and what can we do to engage and support the local community, artists, researchers where the conference is hosted?

3.3.2. Organising Local NIME Chapters and Hubs in LATAM

As well as conferences, another possibility is to host local NIME chapters in LATAM, that is, local satellite conferences and/or NIME-related events that may happen synchronously and or asynchronously with NIME. These chapters may be governed by local NIME committees and/or self-organised by local communities. Chapters may have their own proceedings (e.g., SIGCHI Local Chapters⁴) or might be events where a local hub can remotely stream talks from the conference or watch their recordings and then have a remote live Q&A session with the authors (e.g., IIT Bombay's 'Best of CHI' event⁵, a remarkable example of a sustainable satellite event engaging the local community). These hubs may also run their own workshops, performances and other community-engaging activities throughout the year.

3.3.3. Writing NIME Papers in the Languages of LATAM

Although English may be considered the *lingua franca* in most academic disciplines, and most journals and conferences require submissions to be written in this language, it also imposes a linguistic barrier to academic efforts outside of the *anglosphere*. This barrier is also socioeconomic as not everyone has access to English language education, and it will always represent a disadvantage for communication and expression to English-as-a-second-language (ESL) speakers. How can we lower these barriers for emerging NIME research in LATAM whilst also including their work in the wider corpus of knowledge of NIME? An alternative would be to allow participants to submit and present their work in their native language, especially if the conference is held in a place where most people attending and residing there speak that language. Moreover, it could represent a more amicable entry point for non-English speakers, and their written works could be translated and/or simultaneously interpreted during the conference.

4. Expected Outcomes

Although we aim to have an open-ended and discussion-centred workshop, we expect a series of actionable outcomes that could emerge from it, and that would constitute the first building blocks of a future LATAM NIME community, such as:

- Communication platforms for the community (such as a mailing list, a Discord/Slack Workspace, a Telegram/WhatsApp group, a Forum, etc.)
- Define what can be hosted in the subdomain *latam.nime.org* (such as repositories for NIME related projects, biographies of the members, a blog, tutorials, interviews, etc.)
- Regular events that can gather locally or remotely the community (such as a local NIME conference chapter, regular meetings, festivals, conferences, residencies, etc.)
- Creating a local NIME-related publication (such as a journal, magazine, conference proceedings, etc.)
- Creative projects to foster a LATAM NIME Community (such as creating virtual space for meeting and sharing our contents, original artworks, apps, or platforms to connect the members of the community in different ways, etc.)

We also acknowledge that many other outcomes, beyond the ones stated above, may emerge from the workshop. We look forward to the future of this community and how it may extend beyond NIME, opening to other communities in LATAM, as well as stirring policy changes to support the development of our local communities.

References

- [1] Martin Matus Lerner. Latin american NIMEs: Electronic musical instruments and experimental sound devices in the twentieth century. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 228–233, Porto Alegre, Brazil, June 2019. UFRGS.
- [2] Lauren Hayes and Adnan Marquez-Borbon. Nuanced and Interrelated Mediations and Exigencies (NIME): Addressing the Prevailing Political and Epistemological Crises. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 428–433. Zenodo, June 2020.
- [3] Stefano Fasciani and Jackson Goode. 20 nimes: Twenty years of new interfaces for musical expression. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Shanghai, China, June 2021. NYU Shanghai.
- [4] Adriana Alvarado Garcia, Karla Badillo-Urquiola, Mayra D. Barrera Machuca, Franceli L. Cibrian, Marianela Cioffi Felice, Laura S. Gaytán-Lugo, Diego Gómez-Zará, Carla F. Griggio, Monica Perusquia-Hernandez, Soraia Silva-Prietch, Carlos E. Tejada, and Marisol Wong-Villacres. Fostering hci research in, by, and for latin america. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*, CHI EA '20, page 1–4, New York, NY, USA, 2020. Association for Computing Machinery.
- [5] Juan Pablo Martinez Avila, Chris Greenhalgh, Adrian Hazard, Steve Benford, and Alan Chamberlain. *Encumbered Interaction: A Study of Musicians Preparing to Perform*, page 1–13. Association for Computing Machinery, New York, NY, USA, 2019.
- [6] Filipe Calegario, Marcelo Wanderley, João Tragtenberg, Eduardo Meneses, Johnty Wang, John Sullivan, Ivan Franco, Mathias S Kirkegaard, Mathias Bredholt, and Josh

⁴sigchi.org/chapters/

⁵indiahci.org/best-of-chi2019/

- Rohs. Probatio 1.0: collaborative development of a toolkit for functional DMI prototypes. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 285–290. Zenodo, June 2020.
- [7] Filipe Calegario, Marcelo M. Wanderley, Stéphane Huot, Giordano Cabral, and Geber Ramalho. A method and toolkit for digital musical instruments: Generating ideas and prototypes. *IEEE MultiMedia*, 24(1):63–71, 2017.
- [8] Martin Matus Lerner. Osiris: a liquid based digital musical instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 52–55. Zenodo, June 2017.
- [9] Laurel S Pardue, Miguel Ortiz, Maarten van Walstijn, Paul Stapleton, and Matthew Rodger. Vodhrán: collaborative design for evolving a physical model and interface into a proto-instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 523–524. Zenodo, June 2020.
- [10] Juan Mariano Ramos. Eolos: a wireless MIDI wind controller. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 303–306, Porto Alegre, Brazil, June 2019. UFRGS.
- [11] Matthew Rodger, Paul Stapleton, Maarten van Walstijn, Miguel Ortiz, and Laurel S Pardue. What Makes a Good Musical Instrument? A Matter of Processes, Ecologies and Specificities. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 405–410. Zenodo, June 2020.
- [12] Teodoro Dannemann. Luthiers del siglo xxi, 2019.
- [13] Alex Michael Lucas, Miguel Ortiz, and Dr. Franziska Schroeder. Bespoke design for inclusive music: The challenges of evaluation. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 105–109, Porto Alegre, Brazil, June 2019. UFRGS.
- [14] Miguel A. Ortiz Pérez, Benjamin Knapp, and Michael Alcorn. Diamair: Composing for choir and integral music controller. In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, NIME '07, page 289–292, New York, NY, USA, 2007. Association for Computing Machinery.
- [15] Javier Jaimovich and Francisca Morand. Shaping the biology of emotion: Emovere, an interactive performance. *International Journal of Performance Arts and Digital Media*, 15(1):35–52, 2019.
- [16] Ximena Alarcón Díaz. Conceptual design for intimal: a physical/virtual embodied system for relational listening. *The Journal of Somaesthetics*, 4(2), 2019.
- [17] Juan P Martinez Avila, Vasiliki Tsaknaki, Pavel Karpashevich, Charles Windlin, Niklas Valenti, Kristina Höök, Andrew McPherson, and Steve Benford. Soma design for nime. In Romain Michon and Franziska Schroeder, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 489–494, Birmingham, UK, July 2020. Birmingham City University.
- [18] Isabela Corintha Almeida, Giordano Cabral, and Professor Gilberto Bernardes Almeida. AMIGO: An Assistive Musical Instrument to Engage, Create and Learn Music. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 168–169. Zenodo, June 2019.
- [19] Isabela Corintha and Giordano Cabral. Improvised sound-making within musical apprenticeship and enactivism: An intersection between the 4e's model and dmis. *International Conference on New Interfaces for Musical Expression*, 4 2021. <https://nime.pubpub.org/pub/e4lsrn6c>.
- [20] Adnan Marquez-Borbon and Juan Pablo Martinez-Avila. The Problem of DMI Adoption and Longevity: Envisioning a NIME Performance Pedagogy. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 190–195. Zenodo, June 2018.
- [21] Adnan Marquez-Borbon. Collaborative Learning with Interactive Music Systems. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 581–586. Zenodo, June 2020.
- [22] 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. 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.
- [23] Juan Mariano Ramos. Bandoneon 2.0: desarrollo del prototipo de Bandoneón Electrónico ALFA. In UFSCar, editor, *Anais das XXVII Jornadas de Jovens Pesquisadores. A Ciência e a Tecnologia na Produção de Inovação e Transformação Social.*, São Carlos, Brazil, October 2019. UFSCar.
- [24] Laddy Patricia Cadavid Hinojosa. Electronic _khipu_: Thinking in experimental sound from an ancestral andean interface. *Computer Music Journal*, 44(2-3):39–54, 07 2021.
- [25] Laddy P Cadavid. Knotting the memory//encoding the khipu_: Reuse of an ancient andean device as a nime. In Romain Michon and Franziska Schroeder, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 495–498, Birmingham, UK, jul 2020. Birmingham City University.
- [26] João Tragtenberg, Filipe Calegario, Giordano Cabral, and Geber Ramalho. Tumtá and pisada: Two foot-controlled digital dance and music instruments inspired by popular brazilian traditions. In *Anais do XVII Simpósio Brasileiro de Computação Musical*, pages 76–83, Porto Alegre, RS, Brasil, 2019. SBC.
- [27] João Tragtenberg, Gabriel Albuquerque, and Filipe Calegario. Gambiarra and techno-vernacular creativity in nime research. *International Conference on New Interfaces for Musical Expression*, 4 2021. <https://nime.pubpub.org/pub/aqm27581>.
- [28] Ximena Alarcon-Diaz, Victor Evaristo Gonzalez-Sanchez, and Cagri Erdem. PICO: Intimal: Walking to feel place, breathing to feel presence. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 355–360, Porto Alegre, Brazil, June 2019. UFRGS.
- [29] Ximena Alarcon Diaz, Paul Boddie, Cagri Erdem, Eigil Aandahl, Elias Sukken Andersen, Eirik Dahl, Mari Lesteborg, and Alexander Refsum Jensenius. Sensing place and presence in an intimal long-distance improvisation. *Journal of Network Music and Arts*, 1(1):3, 2019.

Demos

Música Móvel - Criação sonora para web

Cristiano Figueiró¹, Letícia Mayni¹, Leonardo Almeida¹

¹ IHAC, Universidade Federal da Bahia, Salvador, BA.

cristianofigueiro@ufba.br, leticiamayni1@gmail.com, leonardo_almeida1995@hotmail.com

Abstract

In this communication we will present the stage of development of the project "Música Móvel"[1] which proposes the development of an online music creation platform connecting musicians, programmers and those interested in immersive musical experiences in a network[2][3]. The platform runs in the browser and is being developed in a modular way using the combination of javascript[5] with HTML5 and CSS, using functions on top of the Tone.js, P5.js[4] and WebPd (Pure data) libraries. We will demonstrate the general structure of the platform and its characteristics and some functional prototypes of sequencers and synthesizers that we are developing in this phase of the project.

Resumo

Nessa comunicação iremos apresentar o estágio de desenvolvimento do projeto "Música Móvel" que propõe o desenvolvimento de uma plataforma de criação musical online conectando músicos, programadores e interessados em experiências musicais imersivas em rede. A plataforma roda no navegador e está sendo desenvolvida de maneira modular usando a combinação de javascript com HTML5 e CSS, usando funções em cima das bibliotecas Tone.js, P5.js e WebPd (Pure data). Iremos demonstrar a estrutura geral da plataforma e suas características e alguns protótipos funcionais de sequenciadores e sintetizadores que estamos desenvolvendo nessa fase do projeto.

References

- [1] C. Figueiró and G. Soares and B. Rohde. "Música Móvel Apps - Experimentation and open design for musical instruments in Android". In: 5th International Pure data Convention, 2016, Nova York. 5th PdCon Proceedings, 2016.
- [2] Figueiró, C. S (Org.). Desobediência sonora: selos de música experimental e suas tecnologias de sustentabilidade. 1. ed. Salvador: Edufba, 2019. v. 1. 314p.
- [3] C. Figueiró and G. Soares and B. Rohde. ESMERIL - An interactive audio player and composition system for collaborative experimental music netlabels. In: International Conference on New Interfaces for Musical Expression (NIME), 2019, Porto Alegre. Proceedings of

the International Conference on New Interfaces for Musical Expression. Porto Alegre: UFRGS, 2019. p. 170-173.

[4] L. McCarthy and C. Reas and B. Fry. Make: Getting Started with p5.js. San Francisco: Maker Media, 2016.

[5] W. Turner and S. Leonard. JavaScript for sound artists : learn to code with the Web Audio API . Boca Raton : Taylor & Francis, CRC Press, 2017.

Github link:

https://github.com/cristianofigo/musica_movel_web

ContraMonumentos | CounterMonuments

Andreia Oliveira¹, Davi Carvalho¹, Tulio Chiodi¹, Matheus dos Santos Camargo¹, Wagner Antonio¹,
Camila dos Santos¹,
Cristiano Figueiró²

¹ Labinter - Departamento de Artes Visuais, Universidade Federal de Santa Maria, Santa Maria, RS

² IHAC – Universidade Federal da Bahia, Salvador, RS

andreaoliveira.br@gmail.com, davi.carvalho@eac.ufsm.br, tuliochiodi@gmail.com, msc.inspire@gmail.com,
calixtob@gmail.com, mitanoula@yahoo.com.br, cristianofigueiro@ufba.br

Abstract

CounterMonuments | CounterMonuments seeks to problematize, in an artistic proposal with virtual reality and artificial intelligence, traditional monuments from the proposition of other possible worlds that intersect[1]. By implementing artificial intelligence in the proposal, investigating connected/disconnected emotions of life in a COVID 19 world. Emotions are analyzed and classified by an AI that provides data for the Touch Designer software to alter images and audio, related to each emotion. Two deep learning architectures perform in the task of identifying emotions in speech audios for different languages: TCNs and YAMNet. This proposal is carried out by LabInter at the Federal University of Santa Maria/Brazil, by itdLab at Durban University of Technology/South Africa, LAD/UFMS, NANO/UFRJ, IHAC/UFBA and CSGames/PUC/SP. Part of the inter-laboratory exchanges take place via the Internet and virtual networks, such as e-mail exchanges, file sharing on cloud drives, messages via WhatsApp application, videoconferences and, at times, face-to-face exchanges between artists/researchers in the laboratories involved. At first, the speeches are sent for processing by the prototype agent, which is on the event's server, in order to share such data between laboratories. From this online arsenal, the laboratories involved in the proposal performed visually and soundly during the event, using the TouchDesigner software. On the SANSAR platform, CM composes worlds in virtual reality, aiming to explore immersive and interactive possibilities. In this video demonstration, we show a small excerpt of the neural network classifying the emotions of an audio recorded with excerpts in Portuguese and Spanish. Thus demonstrating a poetic use of this technique in the flow of collective creation of the work.

Resumo

ContraMonumentos | CounterMonuments busca problematizar, em uma proposta artística com realidade virtual e inteligência artificial, os monumentos tradicionais a partir da proposição de outros mundos possíveis que se entrecruzam. Ao implementar inteligência artificial na proposta, investigando emoções

conectadas/desconectadas da vida em um mundo COVID 19. As emoções são analisadas e classificadas por uma IA que fornece dados para o software Touch Designer alterar imagens e áudios, relacionados com cada emoção. Há uma performance de duas arquiteturas de aprendizado profundo na tarefa de identificação de emoções em áudios de fala para diferentes idiomas: TCNs e YAMNet. Esta proposta é realizada pelo LabInter na Universidade Federal de Santa Maria/Brasil, pelo itdLab na Durban University of Technology/África do Sul, LAD/UFMS, NANO/UFRJ, IHAC/UFBA e CSGames/PUC/SP. Parte das trocas inter-laboratoriais acontece via Internet e redes virtuais, como trocas de e-mails, compartilhamentos de arquivos em drives de nuvens, mensagens via aplicativo WhatsApp, videoconferências e, ainda, em alguns momentos, intercâmbio presencial entre artistas/pesquisadores dos laboratórios envolvidos. Em um primeiro momento os áudios de falas são enviados para processamento do agente prototipado, que se encontra em servidor do evento, a fim de serem compartilhados tais dados entre laboratórios. A partir desse arsenal online, os laboratórios envolvidos na proposta performaram visualmente e sonoramente durante o evento, utilizando o software TouchDesigner. Na plataforma SANSAR, CM compõe mundos em realidade virtual, visando explorar possibilidades imersivas e interativas. Nesse vídeo demonstração mostramos um pequeno trecho da rede neural classificando as emoções de um áudio gravado com trechos em português e espanhol. Demonstrando assim uma utilização poética dessa técnica no fluxo da criação coletiva da obra.

Github

repository: <https://github.com/davircarvalho/ASEmotion>

References

[1] A. Oliveira and M. Camargo and B. Souza and C. Santos and D. Carvalho and F. Duarte and H. Hildebrand and L. Zindela and M. Szafir and N. Singh. (2021). ContraMonumentos em ambientes imersivos e em redes neurais com classificação de emoções. In: VIII Simposio internacional de innovación en medios interactivos. València/España.

Authors

A

Adnan Marquez-Borbon, 297
Adriano Monteiro, 172
Alessandra Aleluia, 241
Alexandre Silva, 272
Alexandre Silva de Almeida, 249
Alexandre Torres Porres, 229
Aluizio Barbosa de Oliveira Neto, 272
Ana Clara Medina Fonseca, 266
Andre Giolito, 259
Andreia Oliveira, 306
Angelo Cesar Mendes Silva, 8, 37, 60
Arthur Nicholas dos Santos, 209
Augusto Piato Amstalden, 109
Avner Maximiliano de Paulo, 266
Avner Maximiliano Paulo, 233

B

Bruna C. Melo, 152
Bruno Sanches Masiero, 209

C

Caio Campos, 74
Camila dos Santos, 306
Carlos Eduardo Oliveira de Souza, 237, 266
Carlos Eduardo Thomaz, 213
Carlos Guatimosim, 88
Cláudio Gomes, 197, 225, 253, 280, 289
Cleisson José Dias da Silva, 266
Cristiano Figueiró, 304, 306

D

Danilo B. Seufitelli, 152
Davi Alves Mota, 272
Davi Carvalho, 306
Diego Furtado Silva, 8, 37, 60
Douglas Silva, 253
Douglas Sousa, 225, 280

E

Edelson Henrique Constantino, 95
Edimilson Batista dos Santos, 1, 221
Eduardo Bonachela, 259
Eduardo Santos Silva, 136
Emerson Junio Silva Costa, 266
Esteban Viveros Astorga, 229
Estela Ribeiro, 213

F

Fábio dos Passos Carvalho, 266
Felipe Rodrigues, 225, 253, 280
Felipe V. de Almeida, 217
Felippe Brandão Barros, 272
Filipe Calegario, 144, 297
Flaviano Dias Fontes, 166
Flávio Luiz Schiavoni, 1, 233, 237, 245, 249, 266, 272

G

Gabriel Lemos, 259
Gabriel Lopes Rocha, 245, 266
Gabriel P. Oliveira, 152
Gabriel R. G. Barbosa, 152
Gabriel Rodrigues Chaves Carneiro, 266
Geber Lisboa Ramalho, 166
Giordano Cabral, 241
Giordano Ribeiro Eulalio Cabral, 30, 50, 103, 114, 129, 136, 144, 160, 166
Guilherme Feulo do Espírito Santo, 82
Guilherme Ribeiro Eulálio Cabral, 103
Gustavo Carneiro, 293
Gustavo Machado Oliveira, 272
Gutenberg Marques, 205

H

Horhanna Almeida, 293
Horhanna Almeida de Oliveira, 30, 114, 129
Hugo Solis, 297

I

Igino de Oliveira Silva Júnior, 266
Igor Alves Estevam Andrade, 266
Isabela Corintha, 160, 297
Isadora Franco Oliveira, 266

J

Jáder Abreu, 129
Jarbas Jácome, 293
Javier Jaimovich, 297
Jean-Pierre Briot, 16
Jenaro Augusto Barbosa, 221
João Marcos de Oliveira Lara, 266
João Monnazzi, 259
João Pedro Mendes de Oliveira, 266
João Teixeira Araújo, 266
João Teixeira Araújo, 1
João Tragtenberg, 297
Joelden Lopes, 253
Jônatas Araújo da Silva, 237, 266
Jônatas Manzolli, 82, 95
José Henrique Padovani, 74, 88, 189
José Mauro da Silva, 233
Josemar Gama, 253
Juan Pablo Martinez Avila, 297
Juan Ramos, 297
Juciane Araldi Beltrame, 205
Júlio Guatimosim, 88

K

Karen Gissell Rosero Jácome, 209
Konstantinos Vasilakos, 178

L

Leandro Costalonga, 282
Leonardo Almeida, 304
Leonardo Vilela de Abreu Silva Pereira, 25
Letícia Mayni, 304
Luan Luiz Gonçalves, 266
Lucas Estevão dos Santos, 266
Lucas Zampar, 225, 253, 280
Luís Felipe Ortolan, 60
Luis Fernando Umbelino da Silva, 272
Luiz Delando Santos Moreira Júnior, 103
Luiz Gustavo Colzani Monti Sousa, 233, 266

M

Marco Antonio Santos Fernandes, 213
Marcos da Rosa Garcia, 205
Marcus Vinicius Neves, 282
Mariana O. Silva, 152
Mariana Pereira Lellis, 266
Martin Matus Lerner, 297
Matheus Barros, 205
Matheus de Bonfim Rodrigues Jordão, 266
Matheus dos Santos Camargo, 306
Maurício A. Loureiro, 55
Maurício Alves Loureiro, 272
Mavi Pugliesi, 293
Micael Antunes da Silva, 82
Micael Valterlânio da Silva, 37
Michael L. Bastos, 144
Miguel Ortiz, 297
Mika Kuuskankare, 45
Mila Soares de Oliveira de Souza, 16
Mirella M. Moro, 152
Mychelline Cunha, 129

P

Patricia Cadavid, 297
Paulo Ricardo Viviurka do Carmo, 8
Pedro Nuno de Souza Moura, 16

R

Rafael Alves Soares de Andrade, 237, 266
Rafael Sachetto Oliveira, 249
Ravi Shankar Viana Domingues, 272
Rebeca Lima Soares, 266
Regis Rossi A. Faria, 259
Renato Rodrigues Lisboa, 272
Renato Santos Pereira, 121
Ricardo Marcondes Marcacini, 8, 60
Ricardo Scholz, 30
Ricardo Szram Filho, 37
Ricardo Thomasi, 259
Rodrigo Mendes de Carvalho Pereira, 136
Rodrigo Varejão Andreão, 121
Rômulo Augusto Vieira Costa, 266
Rômulo Vieira, 1, 249
Ronaldo C. Prati, 68
Rute Moura, 30, 114, 129, 293

S

Samuel Rodrigues Rabay, 266
Sandro Luis Freire de Castro Silva, 201
Sara Elene Lobato dos Remedios, 201
Sérgio Freire, 74
Simone Bacellar Leal Ferreira, 201

T

Tairone N. Magalhães, 55
Tairone Nunes Magalhães, 272
Teodoro Dannemann, 297
Thiago de Almeida Magalhães Campolina, 272
Tiago Fernandes Tavares, 25, 109
Tiago Lima Bicalho Cruz, 272
Tulio Chiodi, 306

V

Valter Jorge da Silva, 50
Victor T. Hayashi, 217
Vinicius César Oliveira, 189
Vinicius Garcia, 293

W

Wagner Antonio, 306

X

Ximena Alarcón, 297

Y

Yubiry González, 68

Institutions

A

ABC Federal University - Brazil, 68

C

Centro de Artes e Comunicação/Universidade Federal de Pernambuco - Brazil, 297

Centro Universitário Espírito-Santense - Brazil, 121

Centro Universitário FEI - Brazil, 213

Centro Universitário Franciscano - Brazil, 306

D

DACCORD Music Software - Brazil, 136

Departamento de Artes y Humanidades, Universidad Autónoma de México - México, 297

E

Escola de Artes, Ciências e Humanidades/Universidade de São Paulo - Brazil, 259

F

Facultad de Artes, Universidad Autónoma de Baja California - Mexico, 297

Facultad de Artes, Universidad de Chile - Chile, 297

I

Independent Artist/Researcher, 297

Independent researcher - Finland, 45

Instituto De Ciências Matemáticas e de Computação/Universidade de São Paulo - Brazil, 8, 60

Instituto Federal da Paraíba - Brazil, 205

Instituto Federal de Educação Ciência e Tecnologia do Piauí - Brazil, 129

Instituto Federal de Minas Gerais - Brazil, 272

Instituto Federal do Espírito Santo - Brazil, 121

Instituto Federal do Sertão Pernambucano - Brazil, 205

Interface Cultures, Kunstuniversität Linz - Austria, 297

Istanbul Technical University - Turkey, 178

L

LIP6, 16

M

Media and Arts Technology PhD Programme, Queen Mary University of London - UK, 297

Mixed Reality Lab University of Nottingham, 297

P

Programa Doutoral Multimédia em Educação - Universidade de Aveiro - Portugal, 297

S

Sonic Arts Research Centre, Queen's University Belfast - UK, 297

U

Universidad Nacional de Quilmes - Argentina, 297

Universidade de São Paulo - Brazil, 8, 37, 60, 82, 88, 217, 229, 259

Universidade Estadual de Campinas - Brazil, 25, 82, 95, 109, 189, 209

Universidade Federal da Bahia - Brazil, 304, 306

Universidade Federal da Paraíba - Brazil, 129, 205

Universidade Federal de Goiás - Brazil, 172

Universidade Federal de Minas Gerais - Brazil, 55, 74, 88, 152, 189, 272

Universidade Federal de Pernambuco - Brazil, 30, 50, 103, 114, 129, 136, 144, 160, 166, 241, 293, 297

Universidade Federal de Pernambuco - Portugal, 129

Universidade Federal de Santa Maria - Brazil, 306

Universidade Federal de São Carlos - Brazil, 8, 37, 60

Universidade Federal de São João del-Rei - Brazil, 1, 221, 233, 237, 245, 249, 266, 272

Universidade Federal de Viçosa - Brazil, 1

Universidade Federal do Amapá - Brazil, 197, 225, 253, 280, 289

Universidade Federal do Espírito Santo - Brazil, 282

Universidade Federal do Estado do Rio de Janeiro - Brazil, 16, 201

University of Aveiro - Portugal, 160

Keywords

A

Artificial Intelligence, A-Life and Evolutionary Music Systems,
1, 25, 55, 95, 166, 178, 209, 217, 225, 259, 272,
280, 282, 289, 306

C

Computational Musicology, 30, 45, 82, 201, 272
Computer Music and Creative processes, 30, 88, 95, 109,
114, 136, 144, 172, 178, 189, 205, 237, 259, 266,
280, 282, 289, 293, 304, 306

D

Digital Sound Processing, 68, 74, 88, 121, 129, 209, 213, 217,
259, 266, 272, 293

F

Festival proposal (music performance), 197

M

Movement and Gesture, 241, 245, 253, 266, 297
Music Analysis and Synthesis, 1, 30, 114, 129, 209, 259
Music Expressiveness, 144, 160, 209, 245, 282, 293, 297
Music Information Retrieval, 1, 8, 16, 25, 37, 50, 55, 60, 74,
88, 103, 144, 152, 189, 209, 213, 221, 225, 259,
266, 272, 280, 282, 289
Music Perception, Psychoacoustics, and Cognition, 82, 160,
209, 272, 282
Music, Society, and Technology, 25, 103, 166, 201, 205, 209,
217, 233, 237, 266, 282, 293, 297, 304, 306

O

Other, 68

R

Real-time Interactive Systems, 45, 74, 109, 114, 129, 136,
160, 172, 178, 189, 237, 241, 249, 253, 259, 280,
289, 293, 297

S

Sensors and Multimodal Signal Processing, 109, 253
Software Systems and Languages for Sound and Music, 45,
166, 201, 229, 233, 259, 282

REALIZAÇÃO / SPONSORS

