IMMERSIVE

SOUND AND MUSIC COMPUTING

SMC 2024 / 4–6 July
ESMAE, Porto, Portugal

EDITORIAL TEAM
Guilherme Campos DETI-UA / IEETA
Inês Salselas CESEM / ESMAE-P.Porto, IN2PAST
José Vieira DETI-UA / IEETA
Marco Conceição CESEM / ESMAE-P.Porto, IN2PAST
Nuno Fonseca Sound Particles
Rui Penha CESEM / ESMAE-P.Porto, IN2PAST
Ângela da Ponte CESEM / ESMAE-P.Porto, IN2PAST
Dimitrios Andrikopoulos CESEM / ESMAE-P.Porto, IN2PAST

DESIGN
Joana Gonçalves

PROOF READER
Jorge Teixeira

IMMERSIVE SOUND AND MUSIC COMPUTING

SMC 2024

TECH SHEET // 02 **PAPERS INDEX // 05 - 07**

INTRODUCTION // 08 **COMMITTEES // 10**

ORGANISING // 11 ARTISTIC // 12 SCIENTIFIC // 13

CONFERENCE // 15 **PROGRAM 1ST DAY // 16 - 19**

KEYNOTE DERMOT FURLONG // 20

SESSION PAPER 1 // 23

SESSION PAPER 2 // 73 SESSION PAPER 3 // 133

1ST CONCERT // 197

PROGRAM 2ND DAY // 200 - 202

KEYNOTE FRANCISCA GONÇALVES // 203

SESSION PAPER 4 // 206 SESSION PAPER 5 // 276

2ND CONCERT // 347 INSTALLATION // 351

PROGRAM 3RD DAY // 353 - 356

KEYNOTE NATASHA BARRETT // 357

SESSION PAPER 6 // 360

3RD CONCERT // 419

SESSION PAPER 7 // 422

SESSION PAPER 8 // 491

4TH CONCERT // 577

OTHER PAPERS // 580

AWARDS SMC 2024 // 615

BEST PAPER & BEST MUSIC & SONIC ART // 616

PARTICIPANTS // 617

SMC IN PHOTOS // 624

GUIDING CO-CREATIVE MUSICAL AGENTS THROUGH REAL-TIME FLUTE INSTRUMENTAL PLAYING TECHNIQUE RECOGNITION 24

A GEOGRAPHICALLY DISTRIBUTED PERFORMANCE MEDIATED BY A CO-CREATIVE AGENT OVER THE WEB 32

PERFORMING VOICES IN IMMERSIVE THEATER: KINESTHESIS 2.0 FOR EMBODIED PERFORMANCE. 40

REAL-TIME PIANO ACCOMPANIMENT MODEL TRAINED ON AND EVALUATED ACCORDING TO HUMAN ENSEMBLE CHARACTERISTICS 44

REAL-TIME FUTURE-RHYTHM VISUALIZER FOR DJ PERFORMANCE 51

SMUG-EXPLAIN: A FRAMEWORK FOR SYMBOLIC MUSIC GRAPH EXPLANATIONS 57

PATTERNS UI, AN INTERACTIVE TOOL FOR MUSIC EXPLORATION 65

EXPLORING SAMPLING STRATEGIES IN LATENT SPACES FOR MUSIC GENERATION 74

GUITAR CHORD DIAGRAM SUGGESTION FOR WESTERN POPULAR MUSIC 82

A SONIFICATION METHOD FOR MONITORING CHEMICAL SENSOR DATA 89

INTERACTIVE MUSICAL PERIODIC TABLE: SONIFICATION OF VISIBLE ELEMENT EMISSION SPECTRA 97

INTERPRETATIVE DATA SONIFICATION: USING LLMS TO INTERPRET DATA AND GENERATE CONTINUOUS SOUNDSCAPES AT THE SYDNEY OPERA HOUSE 102

ARPEGGIATORUM: AN AUDIO CONTROLLABLE MIDI ARPEGGIATOR 110

SOUNDSCAPE PERSONALISATION AT WORK: DESIGNING AI-ENABLED SOUND TECHNOLOGIES FOR THE WORKPLACE 116

A GENERATIVE FRAMEWORK FOR COMPOSITION-AWARE LOOP RECOMMENDATION IN MUSIC PRODUCTION: DRUM2BASS USE CASE 127

VIBROTACTILE MEMORY: A CASE STUDY OF TIMBRE PERCEPTION TRAINING IN CHILDREN WITH COCHLEAR IMPLANTS USING A VIDEO GAME 134

ASSESSING MUSICAL PREFERENCES OF CHILDREN ON THE AUTISTIC SPECTRUM: IMPLICATIONS FOR THERAPY 142

AN ACCESSIBLE SOFTWARE INTERFACE FOR COLLABORATIVE MUSIC PERFORMANCE 150

SINGING FOR ANXIETY: EXPLORING THE POTENTIALS OF SINGING IN IMMERSIVE VIRTUAL REALITY TO HELP CHILDREN WITH SOCIAL ANXIETY 158

FROM MIDI TO RICH TABLATURES: AN AUTOMATIC GENERATIVE SYSTEM INCORPORATING LEAD GUITARISTS' FINGERING AND STYLISTIC CHOICES 165

I GOT RHYTHM, SO FOLLOW ME MORE: MODELING SCORE-DEPENDENT TIMING SYNCHRONIZATION IN A PIANO DUET 173

SIMULATING PIANO PERFORMANCE MISTAKES FOR MUSIC LEARNING 179

TABLATURE GENERATION FROM LEAD SHEETS FOR FINGER-STYLE SOLO GUITAR 187

EXPLORING UNCHARTED SOUNDSCAPES: INNOVATIVE INTERACTION AND MAPPING TECHNIQUES WITH THE BODYHARP AND THE TEINOPHON 191

QUANTIFYING PITCH DRIFT IN UNACCOMPANIED SOLO SINGING: A COMPUTATIONAL EXAMINATION THROUGH DENSITY-BASED CLUSTERING 207

COMPARING AUDIO BOUNDARY ANNOTATION OF VOCAL POLYPHONY: EXPERTS, NON-EXPERTS, AND ALGORITHMS 212

EXPLORING PERDE-SPACE: 3-D TONNETZ VISUALIZATION FOR MAKAM MUSIC 220

EXPLORING IMMERSION AND SPATIAL AUDIO PERCEPTION IN CONCERTS FOR HEARING-IMPAIRED AND NORMAL HEARING AUDIENCES 225

THE ELUSIVE NATURE OF INATTENTIONAL DEAFNESS: ASSESSING THE INFLUENCE OF VISUAL ATTENTION ON BACKGROUND MUSIC PERCEPTION IN TV PROGRAMMES 234

STATISTICAL ANALYSIS OF MUSICAL FEATURES FOR EMOTIONAL SEMANTIC DIFFERENTIATION IN HUMAN AND AI DATABASES 241

TEMPORAL ANALYSIS OF EMOTION PERCEPTION IN FILM MUSIC: INSIGHTS FROM THE FME-24 DATASET 247

BISAID: BIPOLAR SEMANTIC ADJECTIVES ICONS AND EARCONS DATASET 253

"BACK IN MY DAY...": A PRELIMINARY STUDY ON THE DIFFERENCES IN GENERATIONAL GROUPS PERCEPTION OF MUSICALLY-EVOKED EMOTION 261

EXPLORING THE EMERGENCE OF BEAT INDUCTION USING A 'SWARM OF ONSETS' GENERATIVE MODEL 269

BEAMIX: APPLICATION OF AN AMBISONICS FILTER BANK FOR THE CREATION OF DIRECTIONAL EFFECTS 277

TOWARD A NOVEL SET OF PINNA ANTHROPOMETRIC FEATURES FOR INDIVIDUALIZING HEAD-RELATED TRANSFER FUNCTIONS 285

A SPEAKER AGNOSTIC APPROACH TO SPATIALISATION IN ELECTROACOUSTIC MUSIC 293

CSOUND VS. CHUCK: SOUND GENERATION FOR XR MULTI-AGENT AUDIO SYSTEMS IN THE META QUEST 3 USING THE UNITY GAME ENGINE 301

COMPARING FOUR 360-DEGREE CAMERAS FOR SPATIAL VIDEO RECORDING AND ANALYSIS 309

EXPLORING CONVOLUTIONAL NEURAL NETWORKS FOR MULTIMODAL CLASSIFICATION OF EXPRESSIVE PIANO PERFORMANCE 316

COMBINING HAND POSE ESTIMATION WITH AUDIO FOR NOISE-ROBUST PIANO SCORE FOLLOWING 323

IEMI: AN IMMERSIVE, EVOLVING MOTION-CONTROLLED INSTALLATION 331

A COVER IMAGE COMPOSITION METHOD FOR MUSIC CONTENT USING MULTIMODAL IMAGE RETRIEVAL AND CROPPING TECHNIQUES 339

INTRODUCING MAKAMNETZ – A VIRTUAL GUIDE TO TURKISH MAKAM UNIVERSE 361

MODULE-LEVEL MIDI 367

MIDI 1.5, SO TO SPEAK 375

LIBREMIDI: A CROSS-PLATFORM LIBRARY FOR REAL-TIME MIDI 1 AND 2 383

PROGRAMMING FPGA PLATFORMS FOR REAL-TIME AUDIO SIGNAL PROCESSING IN C++ 389

ENABLING AFFORDABLE AND SCALABLE AUDIO SPATIALIZATION WITH MULTICHANNEL AUDIO EXPANSION BOARDS FOR FPGA 397

PERSPECTIVES ON IMMERSION THROUGH LASER DOPPLER VIBROMETRY 403

CONDITIONING METHODS FOR NEURAL AUDIO EFFECTS 411

HYBRID NEURAL AUDIO EFFECTS 423

IMPLEMENTATION OF A CONTINUOUSLY VARIABLE DELAY LINE BY CROSSFADING BETWEEN SEVERAL TAP DELAYS 431

REAL-TIME PSYCHOACOUSTIC FREQUENCY MASKING COMPENSATION FOR AUDIO SIGNALS WITH OVERLAPPING SPECTRA 439

MODELING THE PULTEC EQP-1A WITH WAVE DIGITAL FILTERS 445

REVITALIZING MULTIMEDIA ART: RESTORATION AND RE-ACTIVATION OF AN AUDIOVISUAL INSTALLATION IN VIRTUAL REALITY 453

MUSIC THEATRE AND DIGITAL ARCHIVES 459

PHASE REPAIR FOR TIME-DOMAIN CONVOLUTIONAL NEURAL NETWORKS IN MUSIC SUPER-RESOLUTION 467

FLOW: A MUTITIMBRAL, POLYPHONIC, MODULAR, ADDITIVE MUSIC SYNTHESIZER 473

LOOM: A SPECULATION OF ORAL TRADITION IN IMMERSIVE MEDIA 481

EXPLORING THE SPACE BETWEEN INSTRUMENT AND CONTROLLER 486

BALANCING PHYSICAL MODELING AND MUSICAL REQUIREMENTS: ALGORITHMICALLY SIMULATING THE CALLS OF HYALESSA MACULATICOLLIS FOR REAL-TIME INSTRUMENTAL CONTROL 492

A UNIVERSAL TOOL FOR GENERATING DATASETS FROM AUDIO EFFECTS 500

STOCHASTIC RESONANCE: MOLDING SOUNDS FROM NOISE 508

SPECULATIVE MACHINE LEARNING IN SOUND SYNTHESIS 515

UNVEILING THE TIMBRE LANDSCAPE: A LAYERED ANALYSIS OF TENOR SAXOPHONE IN RAVE MODELS 523

TEMPO ESTIMATION ON MULTIPLE METRICAL LEVELS WITH SEQUENCY FLUX AND MULTIRESOLUTION ANALYSIS 531

CORRELATIONS BETWEEN OBJECTIVE AND SUBJECTIVE EVALUATIONS OF MUSIC SOURCE SEPARATION 538

RECONSTRUCTING THE CHARLIE PARKER OMNIBOOK USING AN AUDIO-TO-SCORE AUTOMATIC TRANSCRIPTION PIPELINE 546

EXPLORING DEEP LEARNING METHODOLOGIES FOR MUSIC EMOTION RECOGNITION 553

CHORDSYNC: CONFORMER-BASED ALIGNMENT OF CHORD ANNOTATIONS TO MUSIC AUDIO 561

PREVALENCE OF TRESILLO RHYTHM IN CONTEMPORARY POPULAR MUSIC 569

DESIGN AND EVALUATION OF ACCESSIBLE DIGITAL MUSICAL INSTRUMENTS FOR PUPILS WITH NEURODEVELOPMENTAL DISORDERS 581

BIOSENSORS IN IMMERSIVE MUSIC PERFORMANCE: ANDROMEDA FOR FLUTE AND BIOFEEDBACK 589

XTRACK: AUTOMATIC SEGMENTATION OF LIVE MUSIC WITH YAMNET 595

CYBORGUTT: EXPLORING THE BODY'S FLUID BOUNDARIES THROUGH A BIOSENSOR COMPOSITION 601

TRAJECTORY DESCRIPTORS: AN APPROACH TO MUSIC GENRE CLASSIFICATION THROUGH CURVES AND VECTORS IN THE TONNETZ 607

INTRODUCTION

Immersive Sound
Immersive Music
Immersive Computing

Rui Penha

The quest for immersion is — quite literally — all around us, and it has been a significant driver of research and development in the academy, the industry, and within artistic communities in the past decades. But what makes an experience immersive?

Spatial sound quickly comes to mind, and it will be a very relevant part of SMC 2024, both in the paper and the music / sonic art sessions. But there is more to immersion than spatial sound alone:

the timbral possibilities afforded by novel sound design and synthesis algorithms certainly play a big role in the envelopment of a sonic experience (immersive sound);

the temporal development of musical discourse and performance can determine its ability to become, as T. S. Eliot famously wrote, “Music heard so deeply / That is not heard at all, but you are / The music / While the music lasts" (immersive music);

new strategies to engage with digital sound and music, both on studio and on stage, can open new paths for involving human computer interaction (immersive computing).

COMMITTEES

ORGANISING COMMITTEE
ARTISTIC COMMITTEE
SCIENTIFIC COMMITTEE

ORGANISING COMMITTEE

Ângela da Ponte CESEM / ESMAE-P.Porto, IN2PAST
Music & Sonic Art co-chair, Summer School co-chair

Dimitrios Andrikopoulos CESEM / ESMAE-P.Porto, IN2PAST
Music & Sonic Art co-chair

Gilberto Bernardes FEUP / INESC TEC
Summer School co-chair

Guilherme Campos DETI-UA / IEETA
Paper co-chair

Inês Salselas CESEM / ESMAE-P.Porto, IN2PAST
Paper co-chair

José Vieira DETI-UA / IEETA
Paper co-chair

Marco Conceição CESEM / ESMAE-P.Porto, IN2PAST
General chair

Nuno Fonseca Sound Particles
Industry liaison chair

Rui Penha CESEM / ESMAE-P.Porto, IN2PAST
Music & Sonic Art co-chair

ARTISTIC COMMITTEE

Anastasia Georgaki NKUA
Ângela da Ponte CESEM / ESMAE-P.Porto, IN2PAST
Annie Mahtani University of Birmingham
Arshia Samsaminia Aristotle University of Thessaloniki, School of Music studies
Bruno Pereira CESEM / ESMAE-P.Porto, IN2PAST
Carlos Guedes NYU Abu Dhabi
Carlos Lopes Freelance Composer
Dimitri Papageorgiou Aristotle University of Thessaloniki
Dimitris Andrikopoulos CESEM / ESMAE-P.Porto, IN2PAST
Dimitris Maronidis Aristotle University of Thessaloniki
Emma Margetson University of Greenwich
Fani Kosona Music School of Piraeus
Fátima Fonte Guildhall School of Music and Drama
Filipe Lopes CIPEM/INET-md
Francisca Goncalves, FEUP/ INESC-TEC
Henrique Ferreira FEUP
Henrique Portovedo University of Aveiro
Jaime Reis ESML-IPL
Jorge Ramos Independant
Jung In Jung Abertay University
Konstantinos Vasilakos XJTLU
Lorenda Ramou University of Ioannina
Luís Bittencourt Independent Artist-Researcher
Luís Costa Aristotle University of Thessaloniki
Marco Conceicao CESEM / ESMAE-P.Porto, IN2PAST
Michalis Paraskakis Aristotle University Thessaloniki - Music Studies
Miguel Carvalhais Faculty of Fine Arts, University of Porto
Notto Thelle Oslo Metropolitan University
Nuno Peixoto de Pinho CESEM / ESMAE-P.Porto, IN2PAST
Øyvind Brandtsegg NTNU
Orestis Toufektsis University of Music Graz
Øystein Fjeldbo NTNU
Panayiotis Kokoras University of North Texas
Pedro Rebelo Queen's University Belfast
Pedro Pestana Universidade Aberta
Romain Michon Inria
Rui Dias Polytechnic University of Castelo Branco
Rui Penha CESEM / ESMAE-P.Porto, IN2PAST
Scott Wilson University of Birmingham
Telmo Marques CESEM / ESMAE-P.Porto, IN2PAST
Teresa Carrasco Garcia HKB
Yota Morimoto Waseda University

SCIENTIFIC COMMITTEE

Akira Takaoka J. F. Oberlin University
Alain Bonardi CICM
Alex Chechile RISD, SRST
Alexander Carôt Anhalt University of Applied Science
Alexander Refsum Jensenius, University of Oslo
Alexandros Kontogeorgakopoulos NKUA
Andre Holzapfel KTH Royal Institute of Technology in Stockholm
Andrew Horner HKUST
Çağrı Erdem University of Oslo
Carlos Guedes NYU Abu Dhabi
Cat Hope
Dan Overholt Aalborg University Copenhagen
Daniel Hug Zürcher Hochschule der Künste
Davide Fantini University of Milan
Davide Rocchesso Università di Palermo
Davide Andrea Mauro Paderborn University
Diamantino Freitas FEUP
Emanuel Sousa CCG Institute
F. Amilcar Cardoso University Coimbra
Federico Avanzini University of Milan
Federico Simonetta Gran Sasso Science Institute
Florence Leve Université de Picardie Jules Verne - Lab. MIS - Algomus
Francesco Foscarin Johannes Kepler University Linz
Frederico Pereira CCG/ZGDV
Georg Essl University of Wisconsin, Milwaukee
Georgios Marentakis Østfold University College
Gerard Roma University of Huddersfield
Gilberto Bernardes FEUP / INESC TEC
Giorgio Gnecco IMT School for Advanced Studies Lucca
Giorgio Presti Università degli Studi di Milano
Giovanni Santini Xi'an Jiaotong - Liverpool
Grigore Burloiu UNATC
Guilherme Campos DETI - UA
Hanna Järveläinen Zurich University of the Arts
Hans Lindetorp KMH
Haruhiro Katayose Kwansei Gakuin Uniersity
Hiroki Nishino Kochi University of Technology
Inês Salselas CESEM / ESMAE-P.Porto, IN2PAST
Jackson Loth Queen Mary University of London
Jean-Louis Giavitto STMS, CNRS - IRCAM, Sorbonne Université
João Svidzinski CICM

Johan Pauwels Queen Mary University of London
Kjetil Falkenberg KTH Sound and Music Computing Group
Leonardo Fierro Aalto University
Luca Andrea Ludovico University of Milan
Luna Valentim CCRMA
Marcelo Queiroz USP
Marco Conceição CESEM / ESMAE-P.Porto, IN2PAST
Martin Ljungdahl Eriksson University West
Matthew Davies SiriusXM/Pandora
Maxime Popoff INSA Lyon
Michael Boyd Chatham University
Michael Mulshine Stanford University
Michelle Lou University of California, San Diego Cruz
Niccolo' Pretto Free University of Bozen-Bolzano
Nicola Davanzo Università degli Studi di Milano
Olivier Lartillot RITMO, University of Oslo
Panayiotis Kokoras University of North Texas
Pedro Rebelo Queen's University Belfast
Pedro Pestana Universidade Aberta
Pierre Jouvelot MINES ParisTech
Pierre Alexandre Tremblay University of Huddersfield
Raul Masu CMA, HKUST-GZ
Riccardo Russo University of Bologna
Roger Dannenberg School of Computer Science, Carnegie Mellon University
Romain Michon Inria
Rui Paiva DEI - FCTUC, University of Coimbra
Rui Penha CESEM / ESMAE-P.Porto, IN2PAST
Rumi Hiraga Tsukuba University of Technology
Sebastian Schlecht Aalto University
Shiguang Liu Tianjin University
Stefania Serafin Aalborg
Stefano D'Angelo Orastron
Sylvain Marchand La Rochelle University
Thibaut Carpentier Ircam
Thomas Rushton Inria
Tom Barker Audiedge OU
Vanessa Faschi Università degli Studi di Milano

SMC

CONFERENCE

DAILY PROGRAM
KEYNOTES
SESSION PAPERS

2024

Day 1 4th of july'24

09:00 **Registration** Entrance of ESMAE

10:30 **Opening** Theatre Helena Sá e Costa

11:00 - 13:00 **Paper & Poster/Demo session 1**

Guiding Co-Creative Musical Agents through Real-Time Flute Instrumental Playing Technique Recognition

Marco Fiorini (IRCAM, Sorbonne Université, CNRS)
Nicolas Brochec (Tokyo University of the Arts)

A Geographically Distributed Performance Mediated by a Co-Creative Agent over the Web

Thales P. Pessanha (University of Coimbra, CISUC, DEI)
Pedro Martins (University of Coimbra, CISUC, DEI)
Jônatas Manzolli (University of Campinas, NICS)

Performing Voices in Immersive Theater: Kinesthesis 2.0 for Embodied Performance.

Tilemachos Moussas (University Of Athens)
Anastasia Georgaki (NKUA)
Natalia Kotsani (National Technical University of Athens)

Real-time Piano Accompaniment Model Trained on and Evaluated According to Human Ensemble Characteristics

Kit C Armstrong (National Tsing Hua University)
Tzu-Ching Hung (National Tsing Hua University)
Ji-Xuan Huang (National Tsing Hua University)
Yi-WenLiu (National Tsing Hua University)

Real-time Future-rhythm Visualizer for DJ Performance

Masatoshi Hamanaka (RIKEN)

SMUG-Explain: A Framework for Symbolic Music Graph Explanations

Emmanouil Karystinaios (Johannes Kepler University)
Francesco Foscarin (Johannes Kepler University Linz)
Gerhard Widmer (Johannes Kepler University)

Patterns UI, an Interactive Tool for Music Exploration

Rory Sweeney (University of Galway)
Pushkar Jajoria (University of Ireland)
Danny Diamond (University of Galway)
Mathieu D'Aquin (Université de Lorraine)
James McDermott (University of Galway)

14:30 Keynote Speaker

Enactive Immersion
Dermot Furlong

15:30 - 17:30 **Paper & Poster/Demo session 2**

Exploring Sampling Strategies in Latent Spaces for Music Generation

Nádia Carvalho (University of Porto, Faculty of Engineering & INESC TEC)
Gilberto Bernardes (INESC TEC & University of Porto, Faculty of Engineering)

Guitar Chord Diagram Suggestion for Western Popular Music

Alexandre D'Hooge (Université de Lille)
Louis Bigo (Université de Lille)
Ken Déguernel (CNRS)
Nicolas Martin (Arobas Music)

A Sonification Method for Monitoring Chemical Sensor Data

Yutian Hu (Queen Mary University of London)
Tony Stockman (Queen Mary University of London)
Aleksandar Radu (University of Lincoln)
Ernesto Saiz (Teesside University)
Nick Bryan-Kinns (University of the Arts London)

Interactive Musical Periodic Table: Sonification of Visible Element Emission Spectra

William W Smith (Royal Conservatory The Hague)

Interpretative Data Sonification: Using LLMs to Interpret Data and Generate Continuous Soundscapes at the Sydney Opera House
Rodolfo Ocampo (UNSW)
Oliver R Bown (University of New South Wales)
JustinShave (Uncanny Valley Music)
BrendanWright (Uncanny Valley Music)
Caroline Pegram (Uncanny Valley)

Arpeggiatorum: an Audio Controllable MIDI Arpeggiator

Davide Andrea Mauro (Paderborn University)
Axel Berndt (Paderborn University)

Soundscape Personalisation at Work: Designing AI-Enabled Sound Technologies for the Workplace

Thomas Deacon (University of Surrey)
Arshdeep Singh (University of Surrey)
Gabriel Bibbó (University of Surrey)
Mark D. Plumbley (University of Surrey)

A Generative Framework for Composition-aware Loop Recommendation In Music Production: Drum2Bass Use Case

Xiaowan Yi (Centre for Digital Music, Queen Mary Univeristy of London)
Mathieu Barthet (Queen Mary University of London)

17:30 - 19H30 **Paper & Poster/Demo session 3**

Vibrotactile Memory: a Case Study of Timbre Perception Training in Children with Cochlear Implants Using a Video Game
Francesco Ganis (Aalborg University)
Ali Adjorlu (Aalborg University Copenhagen)
Lone Marianne Percy-Smith (Rigshospitalet - Copenhagen University Hosptial)
Cecilia Fernandez Samar (Rigshospitalet - Copenhagen University Hosptial)
Stefania Serafin (Aalborg)

Assessing Musical Preferences of Children on the Autistic Spectrum: Implications for Therapy

Natália Isabel Santos (University of Porto, Faculty of Engineering & INESC TEC)
Gilberto Bernardes (INESC TEC & University of Porto, Faculty of Engineering)
Rita Alcântara (University Center of the State of Pará-CESUP)
Nilzabeth Coelho (University Center of the State of Pará-CESUP)
Alessandra Baganha (University Center of the State of Pará-CESUP)

An Accessible Software Interface for Collaborative Music Performance

Vanessa Faschi (Università degli Studi di Milano)
Luca Andrea Ludovico (University of Milan)
Federico Avanzini (Università degli Studi di Milano)
Emanuele Parravicini (Audio Modeling)
Manuele Maestri (Musica Senza Confini)

Singing for Anxiety: Exploring the Potentials of Singing in Immersive Virtual Reality to Help Children with Social Anxiety

Ali Adjorlu (Aalborg University Copenhagen)
Stefania Serafin (Aalborg)

From MIDI to Rich Tablatures: an Automatic Generative System Incorporating Lead Guitarists' Fingering and Stylistic Choices

Pierluigi Bontempi (University of Padua)
Daniele Manerba (University of Brescia)
Alexandre D'Hooge (Université de Lille)
Sergio Canazza (University of Padua)

I Got Rhythm, so Follow Me More: Modeling Score-dependent Timing Synchronization in a Piano Duet

Akira Maezawa (Yamaha Corporation)

Simulating Piano Performance Mistakes for Music Learning Contexts

Alia Morsi (Universitat Pompeu Fabra)
Huan Zhang (Queen Mary University of London)
Akira Maezawa (Yamaha Corporation)
Simon Dixon (Queen Mary University of London)
Xavier Serra (Universitat Pompeu Fabra)

Tablature Generation from Lead Sheets for Finger-style Solo Guitar

Shunsuke Sakai (Nihon University)
Hinata Segawa (Nihon University)
Tetsuro Kitahara (Nihon University)

Exploring Uncharted Soundscapes: Innovative Interaction and Mapping Techniques with the Bodyharp and the Teinophon

Leo Fogadić (Aalborg University)
Doga Cavdir (Aalborg University in Copenhagen)
Dan Overholt (Aalborg University Copenhagen)

19H30 **Concert session 1**

Terminal

Leonie Strecker (University of Music and Performing Arts Graz)

Travelling without moving

Stefano Catena (De Montfort University)

Too waste, too fast

Francesco Perissi (falta)

Matters 8

Daniel Mayer (Institute of Electronic Music and Acoustics / University of Music and Performing Arts Graz)

KEY NOTE DAY 1

ENACTIVE IMMERSION

DERMOT FURLONG

ENACTIVE IMMERSION

DERMOT FURLONG

Recorded Music is now the most popular form of Music Listening.

Stereo recording has become the accepted commercial norm for audio recording and production. Coincident and Spaced stereo recording techniques were adopted as two standards for stereo recording, each allowing for particular Listener perceptual phenomena to be emphasized.

Coincident stereo capabilities were technologically developed to specify a pan-pot stereo Localisation experience, thereby enabling creative studio recorded music production. Coincident stereo was therefore accepted as an artistically viable studio recorded music production technique.

Spaced stereo recording presents listeners with a degree of Immersion experience, which is often an aesthetically appealing feature of the reconstructed sound. However, to date a creative Immersion control capability has not been realised for studio production use.

This is an artistic limitation, which contributed to the definition and development of a variety of alternative 'Surround Sound' formats over the decades for spatial experience extension.

The question of how 2-channel stereo recordings could be technologically processed to facilitate the artistic manipulation of Listener Immersive experience is offered new answers by the adoption of the Enaction worldview. Enaction demands a focus on the required sensory signal characteristics which enable auditory Immersion to be experienced. These enabling signal characteristics permit the operation of the mental Image Schemas (Organizing Ideas) that are relevant for the perception of Immersion.

Enactive processing thereby permits effective control of varying degrees of perceptual Immersion based on manipulation of standard 2 channel stereo format signals.

DERMOT FURLONG
studied Engineering Science in Trinity College Dublin with a particular personal focus on electronics, acoustics, and audio engineering. He also studied classical guitar and developed an interest in musical acoustics and recording techniques. He subsequently worked as a corporate engineer for Westinghouse Electric in the U.S. and Ireland before returning to Trinity as a lecturer so that he could engage in further research on audio and music related issues. His area of particular interest is that of architectural acoustics, spatial audio perception and its significance for music recording technique. In 1995 he was involved in the definition of the postgraduate Music and Media Technologies (MMT) programme in Trinity, for which he has been Course Director since its initiation in 1996. His current research interests relate to the development of audio engineering technologies derived from musical and cognitive concerns. He has been a member of the Audio Engineering Society, the Acoustical Society of America, and EUCognition.

PAPER & POSTER /DEMO SESSION 1

/ GUIDING CO-CREATIVE MUSICAL AGENTS THROUGH REAL-TIME FLUTE INSTRUMENTAL PLAYING TECHNIQUE RECOGNITION

/ A GEOGRAPHICALLY DISTRIBUTED PERFORMANCE MEDIATED BY A CO-CREATIVE AGENT OVER THE WEB

/ PERFORMING VOICES IN IMMERSIVE THEATER: KINESTHESIS 2.0 FOR EMBODIED PERFORMANCE.

/ REAL-TIME PIANO ACCOMPANIMENT MODEL TRAINED ON AND EVALUATED ACCORDING TO HUMAN ENSEMBLE CHARACTERISTICS

/ REAL-TIME FUTURE-RHYTHM VISUALIZER FOR DJ PERFORMANCE

/ SMUG-EXPLAIN: A FRAMEWORK FOR SYMBOLIC MUSIC GRAPH EXPLANATIONS

/ PATTERNS UI, AN INTERACTIVE TOOL FOR MUSIC EXPLORATION

GUIDING CO-CREATIVE MUSICAL AGENTS THROUGH REAL-TIME FLUTE INSTRUMENTAL PLAYING TECHNIQUE RECOGNITION

Marco FIORINI (marco.fiorini@ircam.fr) (0009-0004-0523-2481) [1,*] and
Nicolas BROCHEC (nicolas.brochec@pm.me) (0009-0000-5922-2079) [2,*]

[1] **IRCAM - STMS, CNRS, Sorbonne Université**, Paris, France
[2] **Tokyo University of the Arts**, Tokyo, Japan

[*] *The two authors contributed equally to this work*

ABSTRACT

This paper presents the development of a classifier for real-time flute Instrumental Playing Technique (IPT) recognition. Our classifier, implemented using Convolutional Neural Networks (CNN), shows a solid state-of-the-art accuracy in detecting different flute IPTs, with enough reactivity and steadiness to be accurate in real-time applications. In particular, we use our classifier's output to guide Somax2, a co-creative Corpus-Based Concatenative Synthesis (CBCS) improvisation system developed within the ERC project REACH, of which we are part of. By implementing a new Label Feature in the corpus annotation and in the output selection process of Somax2, we can drive its musical generation through the flute IPT detected by the classifier. Our real-time IPT recognition system offers a new dimension to the Somax2 interaction paradigm, where its artificial agents are now able to responsively engage with the recognized techniques. Contributing to the broader field of human-machine interaction in computer music, our results have potential applications in improvisation, computer-aided composition and new interfaces for musical expression.

1. INTRODUCTION

The paradigm of Corpus-Based Concatenative Synthesis (CBCS) has significantly transformed real-time sonic exploration in live music performances and creative processes. By exploiting extensive sound corpora, musicians can manipulate and concatenate audio segments, leading to the creation of entirely novel music pieces. This process has been successfully exemplified by tools like CataRT [1], now fully integrated into MuBu [2], or FluCoMa [3], presenting comprehensive toolkits for corpus manipulation that offer abundant possibilities for improvisers, sound designers and composers.

Within the domain of computer music, interaction has consistently played a crucial role in performance. Each musician responds to performance gestures, generating a sound influenced not only by the instrument - acoustic or computer - itself but also by the performance space and the performer's interaction with it. In the context of human-machine improvisation, the challenge becomes one of programming an environment with sufficient flexibility to qualify as interactive in response to a performer. Interaction, at any level, signifies a mutually influential process contributing to the ideas of Creative Instrument and Machine Musicianship [4].

The concept of musical agency assumes fundamental importance for this [5]. While the idea of music agents has evolved into a taxonomy of diverse applications [6], in the realms of music improvisation and co-creation, George E. Lewis's Voyager system stands as an early and successful embodiment of this concept [7]. Subsequent innovations such as Omax [8] and contributions from the Raising Co-Creativity in Cyber-Human Musicianship (REACH) project at IRCAM have further refined the notion of co-creativity in cyber-human applications. Introductions of temporal scenarios [9] and high reactivity [10] have breathed new life into the paradigm of co-improvisation and CBCS applications. These systems are, however, based on different models of musical interaction which rarely take into account the timbral aspect of a particular instrument, and can hardly recognise a change in instrumental techniques in real-time and act accordingly.

This work therefore presents an interactive and modular environment in which a particular flute Instrumental Playing Technique (IPT) is identified in real-time and used to guide the co-improvisation of a CBCS-based artificial agent, responding to the technique in question and maintaining stylistic consistency of improvisation thanks to its generative model. We selected the flute as the instrument to focus on because, to our knowledge, no research other than our own in the field of real-time IPT recognition and co-creative improvisation with AI musical agents has yet been conducted on the flute. We chose Somax2 [10] to handle the interactive co-creative part of our research as it is one of the software developed in one of our teams and part of our research area in the REACH project. In addition, although Somax2 possesses efficient responsiveness and immediate interaction capabilities with a real musician, it has no notion of timbre or recognition of instrumental techniques, being tied solely to pitch and chroma descriptors. Our research therefore aims to extend the interaction paradigm provided by its specific model and to explore general pos-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

sibilities of integrating real-time IPT recognition into co-improvisation systems with artificial agents.

This paper is structured as follows: Section 2 briefly summarises existing approaches to the detection of ITPs and describes our proposed model for real-time recognition of flute IPTs. Section 4 describes the integration of our model in Somax2, and Section 5 explains the implementation of the proposed system. Section 3 details the results obtained, Section 6 discusses them and Section 7 presents the conclusions.

2. REAL-TIME IPT RECOGNITION

2.1 Existing Systems

Several studies tackle IPT identification based on principal component analysis [11] and onset detection [12]. These methods need a set of audio descriptors designed only to detect one specific IPT and cannot detect other IPTs. Thus, they are not sufficient for our task. Different research focusing on violin [13], electric guitar [14], Chinese bamboo flute [15], drums [16], cello [17], guitar [18] and flute [19] tackle IPT identification as a multi-class classification task using machine learning algorithms. A few of them have suggested using IPT classification for real-time performance [17–19]. To our knowledge, only the study on the cello [17] has tested their system in real-world situations. However, the proposed system has yet to reach sufficient accuracy and is not ready for practical application.

As mentioned above, MuBu [2] and FluCoMa [3] Max/MSP libraries can perform real-time machine listening. They provide several real-time audio signal analysis tools and machine learning algorithms. However, the available tools do not allow a thorough parametrization of the neural networks. Therefore, these tools are not adapted for our real-time IPT recognition task.

2.2 The Flute Playing Techniques

The flute can produce a chromatic scale thanks to the keys, from medium (C3≈261 Hz) to treble pitch (C6≈2093 Hz), and has a wide range of playing techniques. The sound of the flute is produced by the friction of the air on the mouthpiece and different changes in the velocity and characteristics of the blown air allow playing different techniques [20]. Some playing techniques of the flute are based on articulations, such as the *staccato*, *pizzicato*, and *flutter-tongue* techniques. Some are based on the air flux blown by the performer, such as the *ordinario*, *aeolian*, and *whistle tone* techniques. Some are based on fingering, such as the *multiphonics* and *trill* techniques, whereas some involve singing while blowing, *play and sing*, or hitting the keys hard, *key click* [21].

The flute can play many other techniques, but existing study [19] has shown that some should not be utilized as they strongly confuse classifiers because of their similarity to other techniques (i.e. the *aeolian and ordinario*, *harmonics*, *discolored fingering* techniques among others). Additionally, our preliminary experiments show that, despite the very good overall accuracy, the real-time recognition of short-time duration playing techniques, such as the *key click*, *pizzicato*, *staccato*, and *tongue ram* techniques, is very challenging for the classifier. Thus, for the sake of stability when used with Somax2, we decided not to train our classifier on short-time duration playing techniques. The aim of our real-time IPT recognition system is to identify the different flute playing techniques as audio data in real time. Therefore, we use a total of 7 playing techniques, as shown in Table 1.

aeolian	flatterzunge	multiphonics	ordinario
play and sing	trill	whistle tone	

Table 1. Flute playing techniques in 7 categories.

2.3 Datasets

We use the audio files of the *GFDatabase* [22] to train our model. The database includes five types of recordings (A, B, C, D, and E) from seven microphones placed at different distances from the source. This microphone set has recorded 11 flute playing techniques pitch by pitch within the respective register for each flute IPT. Our preliminary experiments show that using the A, B, C, and D recordings brought better performances than using all of them. As mentioned in 2.2, we do not use the *key click*, *pizzicato*, *staccato*, and *tongue ram* techniques.

To thoroughly assess our classifier, we evaluated it on a separate test dataset (heterogeneous datasets). We used the *FullSOL* sound bank [23] for our test dataset because it has the same flute playing techniques as in the training dataset. To ensure that the flute playing techniques in the *FullSOL* sound bank match those in the training dataset, we removed the unused techniques. The *GFDatabase* merged the highly similar playing techniques into the same categories. For example, minor and major second trills fall under the *trill* category. Similarly, playing a whistle tone with a glissando or not is considered performing a *whistle tone*. Playing and singing at the same pitch or not is considered performing a *play-and-sing* technique. Therefore, we applied this same methodology to the flute playing techniques of *FullSOL*.

2.4 Data format

To prepare our data, we follow an existing methodology [19]. We downsampled the audio files sample rate to 24kHz considering 12kHz (the Nyquist frequency) is sufficient to cover most flute harmonics. We trimmed the silence in the audio files because it is irrelevant. We then sliced the audio files into 15-frame-long sequences (≈ 320ms). The sequences are analyzed with a Log-Mel-Spectrogram (LMS) analysis from the Torchaudio library [24]. The LMS is computed on 128 bins, the FFT window is fixed at 2048 samples, and the hop size is set to 512 samples (≈ 21.3ms). The minimum frequency is set to 150Hz because lower frequencies are irrelevant for the flute. Each sound file is sliced into a maximum number of data samples according to the number of frames used. When the sequence length is shorter than 15 frames, we pad the audio sample with zeros. The data is then normalized.

2.5 Data Augmentations

For audio classification problems, data augmentation is frequently used to compensate for the lack of training data, and previous studies have demonstrated a notable increase in accuracy [25]. It consists of increasing the amount of data by applying several transformations to the data we already have. To do so, we utilized these three audio transformations: pitch shift, Gaussian noise addition, and time stretch.

For the pitch shifting, the original recording tuning is shifted randomly within a range of ± 100 Hz compared to the original tuning of 440 Hz. This is based on the assumption that the instrument's tuning can change between performances. Additionally, it would provide pitches out of the tonal scale, which can help the classifier generalize on unseen data. Gaussian noise is added to each file of the original recording. This is based on the assumption that noise is generated when the microphone signal is amplified. For the time stretching, we modify the audio sample length of $\pm 20\%$. This is based on the assumption that playing techniques can be performed at different speeds.

2.6 Model Architecture

To perform the real-time IPT recognition, we chose to implement a deep Convolutional Neural Network (CNN) architecture since several studies have shown its efficacy for instrument-related audio classification tasks [17,18,26,27]. In the field of IPT automatic recognition, existing research proposed several design strategies to build CNN architectures [17]. We first tested the suggested configurations by adapting their capacity to our data. Nevertheless, our preliminary tests have shown poor accuracy results on our dataset. Therefore, we propose augmenting the capacity of the suggested CNN architectures by adding several layers and selecting the hyper-parameters that increase accuracy. Figure 1 shows a schematic representation of our architecture.

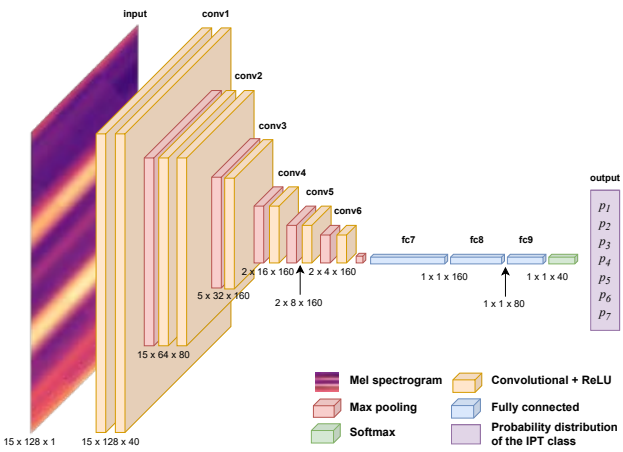


Figure 1. Schematic representation of the proposed architecture.

In our proposed architecture, conv1, conv2, and conv3 use a kernel size of *2x3*, while conv4, conv5, and conv6 use a kernel size of *2x2*. Our preliminary experiments have shown that using a rectangular-shaped kernel size of *2x3* on conv1, conv2, and conv3 led to better accuracy than square-shaped kernel sizes.

We add batch normalization and dropout layers after each convolutional layer. This is because the batch normalization layer fastens the training [28], while the dropout (fixed to 0.25 between each module) prevents overfitting [29]. After the sixth convolutional layer module, we connect three fully connected layers (fc7, fc8, and fc9). We use the identity activation function after these layers because our preliminary experiments have shown better results. Padding is "same" and strides equal to 1 on every convolutional layer.

2.7 Training

The weights of the neural network are initialized with the Xavier Normal Initialization [30]. The training is performed by minimizing the cross-entropy loss function. Mini-batch gradient descent is performed with ADAM optimization with an exponential decay of the learning rate (starting at lr=0.001 with patience of 10 epochs). The classifier is trained for 100 epochs with a V100 GPU machine.

2.8 Measurements

To evaluate the performance of our model, we performed four kinds of measurements. We first measured the accuracy after the training was completed. Then, we computed a confusion matrix to observe how well our classifier identified the playing techniques. These measurements are processed by evaluating the model on successive 15-frame sequences of LMS separated by 15 frames ($\approx$ 320ms).

For a comprehensive assessment of the performances of our model, we conducted a reactivity and steadiness study. For that purpose, we generated a random sequence of two test samples of different playing techniques. The end of the first sample is randomly truncated to create an abrupt change in the IPT. The aim of this test is to determine if the model can detect the change and maintain accurate predictions over time. These measurements are processed by evaluating the model on successive 15-frame sequences of LMS separated by 1 frame ($\approx$ 21.3ms), simulating a real-time situation.

We also measure the total delay time. This was defined by the sum of the audio sample gathering time ($\approx$ 21.3 ms), the audio samples processing time and the time induced by the classifier's prediction [19]. To ensure proper performance, the total delay time should not exceed the time of two frames ($\approx$ 42.6ms). If the system exceeds this limit, the delay will accumulate infinitely. We conducted 10 tests in real-time to measure the audio processing time and classifier-induced prediction time. We then calculated the average and the standard deviation.

3. RESULTS

In this section, we present the results of the evaluation of our model.

3.1 Accuracy and Confusion Matrix

Our model shows a consistent accuracy with a score of 92.56%. The confusion matrix in Figure 2 shows that the model identifies all the playing techniques well. The *trill* technique is the most accurately identified, with an accuracy score of 99.47%. The *multiphonics* technique is the least identified, with an accuracy score of 78.61%.

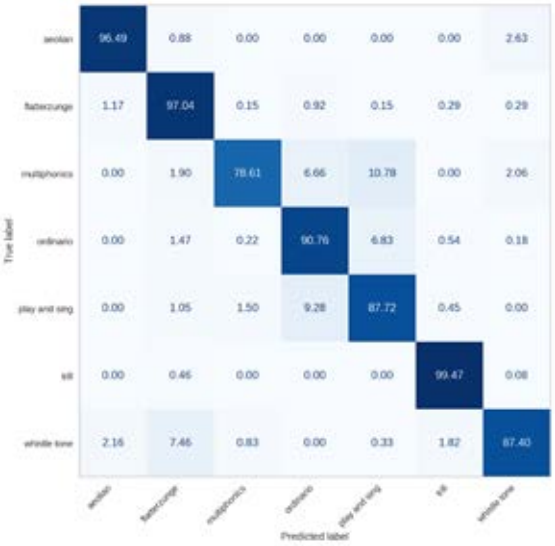


Figure 2. Confusion matrix (%).

3.2 Reactivity and Steadiness Study

We studied how our model reacts and remains steady in its predictions in a real-time situation. The graph in Figure 3 shows that it is reactive to changes and maintains consistency over time. However, two peaks represent misidentifying. The first is between the two techniques, and the second is at the end of the second technique. We think our model may be confused when encountering the first samples of a new technique or when a technique fades out to silence. On this specific test, it had an overall accuracy of 98.66%.

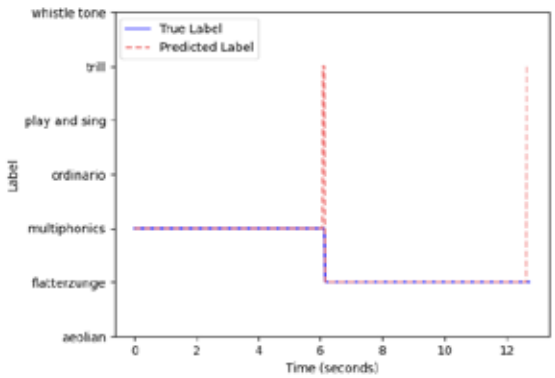


Figure 3. Reactivity and steadiness study: label prediction over time.

3.3 Total Delay Time

To measure the delay time induced by our model and the audio sample processing time, we measured the delay time for each prediction made on the sequence 10 times. We then computed the average and the standard deviation. We found an average of 6.56 ms and a standard deviation of 1.30 ms. With the sample gathering time ($\approx$ 21.3 ms) added, we found a total delay time of 27.86 ms, which is below the limit fixed in 2.8 ($\approx$ 42.6 ms).

4. INTEGRATION WITH SOMAX2

Somax2, developed in the Music Representation team at IRCAM as part of the ERC REACH project, serves to generate stylistically coherent musical interactions, particularly focused on the reactivity aspect of non-idiomatic improvisation. This is achieved through a generative model constructed via statistical learning from a designated corpus in either audio or MIDI format, while interacting with a human improviser. The system, comprehensively detailed in [10], is trained on user-selected musical material to establish a corpus tailored for improvisation purposes. In the following sections, we will give a brief overview of its fundamental concepts and describe the implementation process that resulted in the integration with our real-time IPT recognition system.

4.1 Somax2 Concepts

As described in [10], there are three main processes in the Somax2 system: building and modeling a corpus, influencing, and generating output. Diverging from pure generative methodologies, Somax2 generates improvisational material directly from the original musical data without recourse to an independent model. This genre-agnostic approach involves segmenting the musical corpus into fragments and subjecting each to a multilayer analysis based on various musical features, such as harmony (chroma) and melody (pitch). The resultant multilayer representation forms the foundation of a navigation model, facilitating non-linear exploration of the original data. Thanks to this, Somax2 stands apart from other generative systems by preserving the intrinsic qualities of a specific musical corpus, integrating them directly into the model, and presenting a highly reactive take on generative music tools.

In real-time interactions with a musician, Somax2 performs a similar segmentation and multilayer analysis on the input stream, generating activation peaks in the sequential memory when the analysed input matches the information stored in a specific segment of the corpus. These peaks, resembling probability distributions, signify potential output candidates. When a live input audio or MIDI stream arrives, this is sliced according to either the pitch or the onset and then analysed with pitch and chroma descriptors. The result is a rendition of the input stream in a symbolic domain, called influences, for any of the different layers. Each incoming influence is compared, at a given point in time, with the material stored in the corpus (which was previously built with a similar analysis,

but offline) in order to assess a possible match in the structure of the musical events produced. If this correspondence between internal and external multilayer classification occurs, a trigger peak is generated. At each new time step (after each new event generated) the peaks derived from each layer of the analysis are merged and scaled, depending on whether they are reinforced or not, and at this point the peak with the best score is chosen as the candidate for output.

The interplay of corpus segments, influences and peaks delineates the co-improvisation process. The system's capacity to dynamically balance its internal logic, thanks to self-listening, with external input from the musician results in a coherent yet complex musical output. This co-improvisation process effectively simulates the presence of an active agent in a collaborative musical experience, providing a mixture of coherence and unpredictability, and shows typical aspects of emergence and non-linear dynamics in improvisation, where the apparition of non-linear regimes of structure formation lead to rich musical co-evolution of forms [5].

4.2 Output Selection

Somax2 has up to this point mainly been operating on two musical dimensions when selecting its output: chroma and pitch. In later releases (from Somax 2.5) a number of so called scale actions were introduced to give the user some degree of control over other musical dimensions and various temporal aspects of the generated output. These scale actions operate as gaussian filters by scaling the score of the selected peak according to a parameter $\gamma \in \mathbb{R}_{[0,1]}$. Thanks to this, a peak p, as previously defined in [31], can now be detailed as: $p = \begin{bmatrix} t^{(\mathcal{C})} \\ \gamma y \end{bmatrix}$, where y is a score (height) designated to each peak by the model, $t^{(\mathcal{C})}$ the temporal position of the corresponding slice's onset, and γ can be described as a scale action function operating on a specific trait θ defined in the corpus.

4.2.1 Region Mask

The Region Mask scale action allows the user to specify a region of indices within the corpus that the output should be selected from, i.e. to select only one of multiple sections in a structured corpus. More specifically, the user specifies an interval $[a, b],\ a, b \in \mathbb{Z}_{[1,U]},\ a < b$ where U denotes the size of corpus $\mathcal{C}$, and applies a parameter γ (according to the new peak definition) defined as:

$$\gamma = \begin{cases} 1 & \text{if } a \leq u\left(\mathcal{S}_p^{(C)}\right) \leq b \\ 0 & \text{otherwise.} \end{cases}$$

where $S_p^{(C)}$ denotes the slice in the corpus C corresponding to the time t of the peak p and the function $u(S_p^{(C)})$ returns the index $u \in [1, ..., U]$ of this slice.

In spite of this, the need to have an indefinite number of events corresponding to any instrumental class while being able to efficiently switch from one class to another are missing points in favour of our design idea, detailed in later sections.

4.2.2 General Output Selection Mode

There are other methods in Somax2 to handle the selection of the generated output, but they all rely on filtering and selecting peaks based on the pitch and chroma descriptors the model works with. Since this implementation does not allow working on the detection and handling of different IPTs, we found it necessary to create a label filter that works as a kind of dynamic region filter. The output of this filter we implemented will then make use of the available output selection methods, thus working on the pitch and chroma levels, and the peak selection and decay processes illustrated above.

4.3 Label Feature Formalisation

As previously described in Section 4.2, the output selection modes currently available in Somax2 are not sufficient for integration with our real-time IPT classifier. We have thus defined and implemented a new scale action, called Label Feature, designed to enhance the capabilities of output selection modes in Somax2 when integrated with our system, enabling the user to selectively extract musical segments from a corpus based on the IPT class, resulting in a specific dynamic region masking.

4.3.1 Label Filter Parameter

To enable dynamic region selection based on label classification, we introduce a label filter parameter $\gamma(L)$ defined as:

$$\gamma(L) = \begin{cases} 1 & \text{if } L \in S_L \text{ (filter active)} \\ 0 & \text{otherwise (filter inactive),} \end{cases}$$

where S_L is a user-defined set of labels corresponding to the recognized IPT, determining the criteria for inclusion (and thus providing flexibility to it), and L denotes the classifier output, representing distinct labels corresponding to the recognized IPT, such as staccato, pizzicato, etc.

This filter parameter serves as a binary indicator for inclusion or exclusion of musical slices based on their associated labels. If the detected label L is in the specified set S, the filter is turned on as a switch $\gamma(L) = 1$, allowing slices associated with that label to be included in the output. Otherwise, the switch is off as $\gamma(L) = 0$, excluding slices with labels outside the set S_L.

4.3.2 Label Filter Function

To define a Label Filter, where the label of each slice determines which slices are selected for playback, we can define the set of slices associated with label l as S_l, keeping in mind that L represents the label output by the classifier. The filter can be represented as $F(L) = \bigcup_l S_l$,, where $F(L)$ represents the filtered set of slices based on the classifier output L, and $\cup$ denotes the union operation, which combines all sets S_l corresponding to the labels detected by the classifier.

In particular, giving a set of N possible labels, S_l can be defined for each label l, and the filter function becomes $F(L) = \bigcup_{l=1}^{N} S_l$, where $F(L)$ is the set of all slices that have labels matching the classifier output.

The overall filter function $F(L)$ can then be expressed, considering the filter parameter, as:

$$F(L,\gamma) = \bigcup_{l=1}^{N} \{S_l \mid \gamma(l) = 1\},$$

where S_l represents the set of musical slices associated with label l. In particular, $F(L,\gamma)$ is the union of sets S_l, where each set S_l contains slices from the corpus associated with label l. Only sets S_l for which $\gamma(L) = 1$ are included in the union, ensuring that slices are selected based on the presence of the label in the predefined set S.

At this point, the output slice S_w at time step w is a function of the peak matrix P_w (which contains the peaks combined with previous influences, shifted and decayed corresponding to the time passed since the previous influence according to the regular output selection mode), the corpus C and the history of previous output slices H_w. More specifically, we can define a set $\Xi = \{p_1, ..., p_H\}$ of viable peak candidates with the corresponding set of slices $\xi = \left\{\mathcal{S}_{p_1}^{(\mathcal{C})}, \ldots, \mathcal{S}_{p_H}^{(\mathcal{C})}\right\}$, where $S_{p_h}^{(\mathcal{C})} = F(L,\gamma), \quad \forall \mathcal{S}_{p_h}^{(\mathcal{C})} \in \xi$.

As shown in Figure 4, this formulation allows for the dynamic selection of slices based on the presence of specific N labels l in the predefined set S_l. Thus, the system adapts its output to focus on musical segments characterized by the recognized IPT. This combination of label filter parameter and label filter function results in the new implemented Label Feature.

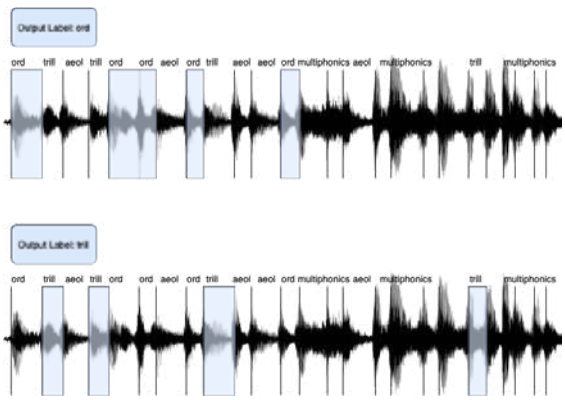


Figure 4. Result of the label filter function on two different IPTs (ord for ordinario, trill) annotated on the same audio corpus. The blue highlights show the output selected by the filter according to the correspondence with the various labels.

5. IMPLEMENTATION

In this section, we describe the implementation of the real-time IPT recognition system with Somax2 improvisation agents, called players. For the real-time IPT recognition system part, we use a Python back-end that runs in the background. The same principle applies for Somax2, where Max/MSP works as a front-end providing audio features, UIs and tools to interact in real-time with the application, but the whole generative core is handled by a Python back-end [10]. Therefore, our implementation ensures architecture consistency and the possibility of integrating our real-time IPT recognition system into the more complex Somax2 architecture.

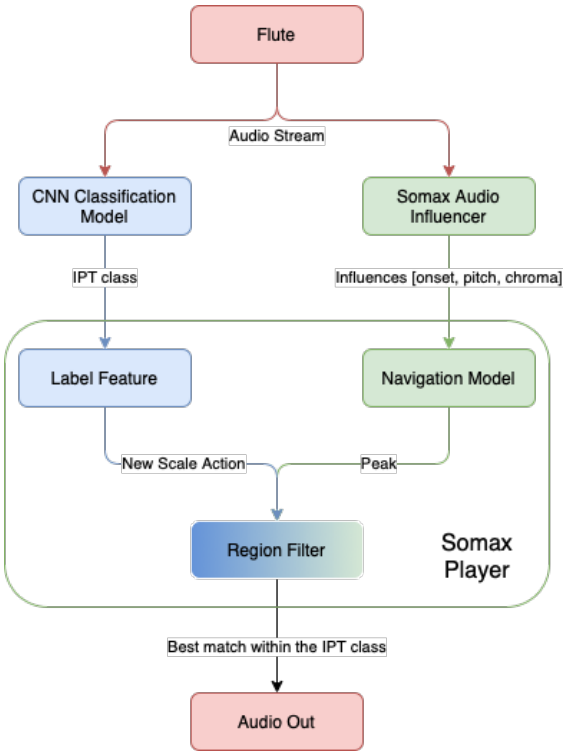


Figure 5. Detailed logic workflow derived from our implementation. The items in red represent the audio elements, those in green are the current modules and objects in the Somax2 model, while the ones in blue represent implementations of our modules.

5.1 Python back-end

On the real-time IPT recognition system side, the Python back-end processes the flute audio stream and runs the model in real time. We describe the processing flow step by step. The incoming audio signal is downsampled to 24 kHz and stored in a 7680 samples-length (15 frames) buffer. The stored samples are analyzed using the LMS before being normalized. Our model predicts the resulting 15 frames-long sequence, and an *argmax* function identifies the class with the highest probability. The class name is sent to Max/MSP through OSC protocol. The last 512 samples are removed, and the buffer is refreshed with the new 512 samples (1 frame). This process repeats. On Somax2 side, since its generative core lies in its Python code, the label feature mentioned in Section 4.3 has been implemented here as a new scale action [10] available to the somax.player.

5.2 Max/MSP front-end

The Max/MSP front-end serves multiple roles: it handles the flute audio stream from the audio interface, sends it to the Python back-end, receives the predicted IPT, and runs Somax2 and its improvisation agents. To send the

audio signal of the flute to Python back-end, we use the BlackHole [1] audio routing device. The predicted IPT is received through the OSC protocol and processed in Somax2 through messages corresponding to the label feature scale action, driving in real-time the selection of the corresponding IPT class. The complete workflow of interaction derived from our implementation can be seen in detail in Figure 5.

6. DISCUSSION

Our model has shown a solid state-of-the-art accuracy score of 92.56%, and the confusion matrix in Figure 2 illustrated its performance by classifying the various flute playing techniques. The reactivity and steadiness study has shown that our model can be accurate in real time. The misidentification issue can be alleviated by implementing a moving average on the probability distribution outputted by the classifier. However, this sacrifices reactivity. The calculation of total delay time showed that our model can be run in real-time without exceeding the limit ($\approx$ 42.6 ms). As mentioned in section 2.2, the short time duration playing techniques such as *key click*, *pizzicato*, *staccato* and *tongue ram* techniques have been removed from the datasets for our experiment. Indeed, our CNN-based classifier model did not provide enough stability for identifying these techniques during preliminary tests. We think adding audio descriptors [32] to Log-Mel-Spectrograms data samples may help the classifier, as they would bring better audio representations.

Future work will include testing and perfecting our system in studio with professional improvisers. We intend to expand the model to other instruments, starting with the clarinet and electric guitar, thus building specific datasets. Thanks to the future conception of a specific Max/MSP object, which would also improve the architectural aspects of the current implementation, the system could also be natively accessible in Max. In this way, user utilisation would also be simplified and a hybrid, multi-level training system could be conceived, in which in addition to providing specific datasets, the user can carry out new training stages on their own database to calibrate the system to their instrument.

7. CONCLUSIONS

This paper presented the application of a state-of-the-art model to recognize flute instrumental playing techniques in real time and to guide the interaction of improvisational artificial agents through this detection, using the co-creative system Somax2.

By exploring playing technique possibilities of a musical instrument, our implementation contributes to the domain of immersive sound, giving new perspectives to work on real-time recognition of IPT and their integration with different human-machine co-creativity frameworks. At the same time, by working on new strategies to engage with sound and music in real time, our research falls within the domain of immersive computing, providing new insights for human-computer interaction in the context of improvised music but potentially extending to computer-assisted-composition scenarios and new interfaces for musical expression.

[1] https://existential.audio/blackhole/

Acknowledgments

This research is supported by the European Research Council (ERC) as part of the Raising Co-creativity in Cyber-Human Musicianship (REACH) Project directed by Gérard Assayag, under the European Union's Horizon 2020 research and innovation program (GA #883313). The authors would like to thank Joakim Borg for the continuous Somax2 expertise.

8. REFERENCES

[1] D. Schwarz, G. Beller, B. Verbrugghe, and S. Britton, "Real-Time Corpus-Based Concatenative Synthesis with CataRT," in *9th International Conference on Digital Audio Effects (DAFx)*, Montreal, Canada, Sep. 2006, pp. 279–282.

[2] N. Schnell, A. Röbel, D. Schwarz, G. Peeters, R. Borghesi *et al.*, "Mubu and friends–assembling tools for content based real-time interactive audio processing in max/msp," in *ICMC*, 2009.

[3] P. A. Tremblay, O. Green, G. Roma, J. Bradbury, T. Moore, J. Hart, and A. Harker, "Fluid corpus manipulation toolbox," Jul. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.6834643

[4] R. Rowe, *Interactive Music Systems: Machine Listening and Composing*. Cambridge, MA, USA: MIT Press, 1992.

[5] D. Borgo, *Sync or Swarm: Improvising Music in a Complex Age*. AC Black, 2005.

[6] K. Tatar and P. Pasquier, "Musical agents: A typology and state of the art towards Musical Metacreation," *Journal of New Music Research*, vol. 48, no. 1, pp. 56–105, 2019.

[7] G. E. Lewis, "Too many notes: Computers, Complexity and Culture in Voyager," *Leonardo Music Journal*, vol. 10, pp. 33–39, 2000.

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

[9] J. Nika, K. Déguernel, A. Chemla, E. Vincent, G. Assayag *et al.*, "DYCI2 agents: merging the "free", "reactive", and "scenario-based" music generation paradigms," in *International Computer Music Conference*. Shangai, China, 2017.

[10] G. Assayag, L. Bonnasse-Gahot, and J. Borg, "Cocreative Interaction: Somax2 and the REACH Project," *Computer Music Journal 1-19*, pp. 1–34, Feb. 2024. [Online]. Available: https://doi.org/10.1162/comj_a_00662

[11] M. Malt and M. Gentilucci, "Real time vowel tremolo detection using low level audio descriptors," *arXiv preprint arXiv:1511.07008*, 2015.

[12] M. Malt and E. Jourdan, "Real-time uses of low level sound descriptors as event detection functions using the max/msp zsa. descriptors library," in *Brazilian Symposium on Computer Music*, vol. 12, 2009, pp. 45–56.

[13] L. Su, H.-M. Lin, and Y.-H. Yang, "Sparse modeling of magnitude and phase-derived spectra for playing technique classification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 2122–2132, 2014.

[14] Y.-P. Chen, L. Su, Y.-H. Yang *et al.*, "Electric guitar playing technique detection in real-world recording based on f0 sequence pattern recognition." in *ISMIR*, 2015, pp. 708–714.

[15] C. Wang, E. Benetos, V. Lostanlen, and E. Chew, "Adaptive scattering transforms for playing technique recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 1407–1421, 2022.

[16] P. Herrera, A. Yeterian, and F. Gouyon, "Automatic classification of drum sounds: a comparison of feature selection methods and classification techniques," in *Music and Artificial Intelligence: Second International Conference, ICMAI 2002 Edinburgh, Scotland, UK, September 12–14, 2002 Proceedings*. Springer, 2002, pp. 69–80.

[17] J.-F. Ducher and P. Esling, "Folded cqt rcnn for real-time recognition of instrument playing techniques," in *International Society for Music Information Retrieval*, 2019.

[18] A. Martelloni, A. P. McPherson, and M. Barthet, "Real-time percussive technique recognition and embedding learning for the acoustic guitar," *arXiv preprint arXiv:2307.07426*, 2023.

[19] N. Brochec and T. Tanaka, "Toward real-time recognition of instrumental playing techniques for mixed music: A preliminary analysis," in *International Computer Music Conference (ICMC 2023)*, 2023.

[20] N. H. Fletcher and T. D. Rossing, *The physics of musical instruments*. Springer Science & Business Media, 2012.

[21] C. Levine and C. Mitropoulos-Bott, *The Techniques of Flute Playing I/Die Spieltechnik der Flöte I*. Bärenreiter-Verlag, 2019.

[22] N. Brochec and W. Howie, "GFDatabase: A Database of Flute Playing Techniques," Feb. 2024. [Online]. Available: https://doi.org/10.5281/zenodo.10612396

[23] 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.

[24] Y.-Y. Yang, M. Hira, Z. Ni, A. Chourdia, A. Astafurov, C. Chen, C.-F. Yeh, C. Puhrsch, D. Pollack, D. Genzel, D. Greenberg, E. Z. Yang, J. Lian, J. Mahadeokar, J. Hwang, J. Chen, P. Goldsborough, P. Roy, S. Narenthiran, S. Watanabe, S. Chintala, V. Quenneville-Bélair, and Y. Shi, "Torchaudio: Building blocks for audio and speech processing," *arXiv preprint arXiv:2110.15018*, 2021.

[25] S. Wei, S. Zou, F. Liao *et al.*, "A comparison on data augmentation methods based on deep learning for audio classification," in *Journal of physics: Conference series*, vol. 1453, no. 1. IOP Publishing, 2020, p. 012085.

[26] V. Lostanlen and C.-E. Cella, "Deep convolutional networks on the pitch spiral for musical instrument recognition," *arXiv preprint arXiv:1605.06644*, 2016.

[27] Y. Han, J. Kim, and K. Lee, "Deep convolutional neural networks for predominant instrument recognition in polyphonic music," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 208–221, 2016.

[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. pmlr, 2015, pp. 448–456.

[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.

[31] J. Borg, "Dynamic Classification Models for Human-Machine Improvisation and Composition," Master's thesis, Aalborg University, 2020.

[32] G. Peeters, "A large set of audio features for sound description (similarity and classification) in the cuidado project," *CUIDADO Ist Project Report*, vol. 54, no. 0, pp. 1–25, 2004.

GEOGRAPHICALLY DISTRIBUTED PERFORMANCES MEDIATED BY AN ARTIFICIAL CO-CREATIVE AGENT

Thales R. P. PESSANHA (0000-0002-3046-9710) [1,2], **Pedro MARTINS** (0000-0002-3630-7034) [2], and **Jônatas MANZOLLI** (0000-0003-4257-7118) [2,3]

[1] *Institute for Interdisciplinary Research*, **University of Coimbra**, *Computational Media Design*, Coimbra, Portugal
[2] *CISUC/LASI, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering*, **University of Coimbra**, Coimbra, Portugal
[3] *Interdisciplinary Nucleus for Sound Studies*, **University of Campinas**, Campinas, Brazil

ABSTRACT

The convergence of digital sound technologies and immersive computing presents a fertile ground for experimentation, enabling artists to craft immersive sonic experiences that push the boundaries of traditional creative processes and promoting the active involvement of both artists and audiences, fostering creative collaborations and deep cognitive and emotional connections. However, in distributed performance settings, immersive environments bring challenges in terms of involvement, since the contexts in which both the performers and the audience are inserted are distinct. This, in turn, requires the development of new strategies and techniques to engage with digital sound, since this artifact is responsible for interconnecting the contexts of such environments. Based on this, we outline the architecture of a system aimed at creating immersive environments for distributed performance using an artificial co-creative agent. Also, we discuss both the computer implementation and performance tests of the DSP calculation to measure its feasibility in real-time performances. Finally, we consider that this architecture opens up new paths for involving human-computer interaction, offering innovative strategies for sound design in distributed performance contexts and the creation of accessible digital sound systems on the web.

1. INTRODUCTION

The emergence of new technologies, marked by the exponential growth in CPU/GPU processing power and the reduction of devices, has opened up new possibilities for integrating immersive technologies in the domains of music and performance art.

These emergent technologies also enabled the rise of systems that allow the connection of a set of technologies and objects in embedded systems, which grant communication and interaction between different elements over the internet [1], and helped to integrate local environmental information through motion sensors, cameras, and microphones, since mobile devices connected to the web create a much more connected environment and supply users with easy access to multimedia content [2].

Also, the specification and implementation of Application Programming Interfaces (APIs), Websocket protocols, synchronous and asynchronous processing, and other paradigms, infrastructures, and toolkit which are integrated into the environment and development of the web applications, have fostered the web platform into a fertile playground for artists and musicians [3], simplifying and democratizing the use of devices for sound diffusion, spatially distributed rendering, and the exploration of both social and sound interactions in their performances and creative projects.

Altogether, this interconnection promoted the active involvement of both artists and audiences, furthering creative collaborations and deep cognitive and emotional connections, drawing on a wealth of perceptual, motor and affective resources, enriching the experiences of all participants involved.

Nevertheless, Networked Music Performances still presents some challenges and drawbacks. These include the requirement for robust technological infrastructure facilitating seamless communication among devices, as well as ensuring connectivity conducive to meaningful performances [1, 4]. Moreover, the scarcity of sophisticated, simple-to-use tools tailored for non-expert users have required both researchers and artists to familiarize themselves with fundamental technologies [3]. Lastly, latency mitigation represents another significant challenge, given musicians' highly sensitivity to interaction delays. In a network performance setting, such delays are not only unavoidable but also have a physical lower limit [5].

In this paper, we propose a system's architecture which, based on management guided by an artificial co-creative agent, enables geographically distributed web-based musical performances. For such, we developed a prototype system called *JANIS* (Joint Audio Networking Interactive System) to devise engagement based on symbolic interactions [6]. This enables both musicians and co-creative agents to interact, learn and progress with each other through a cross-learning process which allows them to share attributes – such as intentionality, relevance, adequacy, aim, discovery and style [6] – and structures in both micro and macro scenarios.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

This discussion contributes for the understanding and development of better strategies to engage and adapt musicians in contexts of distributed performance, enhancing their perceptual aspects and experience in a Networked Music Performance (NMP) context.

The remainder of this paper is organized as follows: in section 2, we provide an explanation about the concepts of co-creativity and the use of symbolic interaction to produce relationships between musicians and artificial agent. Section 3 presents a brief description about current technologies in Networked Music Performance research field. In Section 4 we address the architecture of *JANIS* system, the structure developed to process digital audio signal, which allows artificial co-creative agent's decisions during performance, and some initial results. Finally, Section 5 presents concluding remarks.

2. CO-CREATIVITY AND SYMBOLIC INTERACTION

The definition of concepts and psychological aspects related to creativity is something that has produced an extensive literature ranging from the definition of creativity in the creative process of the arts [7], the creativity of important creators in the history of humanity [8], and the manifestation of creativity as a kind of catalytic energy [9].

In practical terms, we can define creativity as the ability to come up with ideas or artifacts that are new, surprising and valuable [10]. Moreover, as creativity is grounded in everyday abilities – such as conceptual thinking, perception, memory, and reflective self-criticism – it can be defined as an aspect of human intelligence in general [10], entering into virtually every aspect of life and being part, to some extent, of everyone's ability to be creative.

Thus, whether from ideas – such as concepts, poems, musical compositions, scientific theories, culinary recipes, choreography, jokes and so on – or from artifacts – paintings, sculptures, steam engines, vacuum cleaners, pottery, origami, penny whistles and many other things we can name – creativity happens from the correspondence to three types of surprise [10]: by making unfamiliar combinations of familiar ideas, by the exploration of conceptual spaces in people's minds, or by the transformation of these conceptual spaces, which ends up transgressing the space of creation.

In recent years, some theorists, such as Williams [11], define creativity as a dynamic and iterative process that unfolds when individuals actively engage and disengage with a multifaceted array of stimuli – ranging from interactions with fellow individuals to immersion in diverse information, ideas, and the physical environment.

Also, creativity manifests itself through the prism of novelty, utility, and a grounding in reality with discernible goals and developmental progression over time [11], extending beyond a binary perspective, encompassing a continuum that spans from small-c creativity, focusing on problem-solving and incremental improvements, to big-C creativity, which exerts profound influences on cultures.

Crucially, the social dimension emerges as a pivotal element, highlighting creativity as an inherently social activity where interactions with both the social and physical environment shape the creative process [11]. Moreover, the intrinsic nature of creativity positions it as an autotelic pursuit, where the enjoyment derived from the process itself serves as a potent catalyst for producing outcomes of heightened creative merit [11].

When we focus on this perspective of creativity in the field of computer music, we see in recent years the development of interactive systems rooted in the evaluation of historical data, in attention to the ongoing sound environment, and dynamic forecasting strategies spanning multiple time scales, being designed for real-time responsive interactions and forward-looking projections [6].

The combination, therefore, of temporal and spatial elements generated by both the algorithms and the dynamic composition, provides a creative projection space capable of sustaining and constantly reviving the musicians' interest in creating with an artificial agents.

Furthermore, the reciprocal causality fostered by the interaction between agents – whether they are real or artificial – shows us that, from a cognitive point of view, they come together in joint action to show each other the paths of performance [12, 13], with their reciprocal production also being a source of learning for others, that is, their relationships are not causal, but causative [12].

This situation of adaptive interdependence is qualified within the musical domain as *symbolic interaction* [6] and involves every level of musical representation. Whether at the level of signal, structure, real or simulated cognitive modalities, these interactions are distributed between human and artificial agents.

It is from this context, thus, that co-creativity is established, arising emergent processes that, based on intertwined musical contributions from humans and machines, goes beyond the mere sum of their individual capabilities [6]. This collaborative process gives rise to unexpected outcomes, influencing the internal states of the performers through intricate dynamics, resulting in moments of tranquility, gradual progression, or abrupt shifts.

At its core, co-creativity – whether produced by human-human or human-computer interaction – is facilitated by generative learning mechanisms that are both cross-referenced and reflexive, with each agent's input at a given time combining its own productions with those of others, all while incorporating memories of past interactions [6]. This interplay fosters a rich and dynamic creative environment, where human and machine coalesce to produce novel and evolving musical expressions.

The synergy produced by the agents articulates actions and dynamic formations unattainable through individual efforts. The emerging forms evolve in a non-linear way, defying conventional molds and norms. This underlines the ability of co-creativity to not be a mere aggregation of individual contributions, but rather the birth of new and irreducible expressions.

As we discussed in this section, symbolic interactions can provide emergent processes during performance that require adaptation from both human and artificial performers, establishing a process that we can call co-creativity.

This gives rise to elements that, from our cognitive point of view and in accordance with the definitions of creativity described previously, can be characterized as novelty.

In our proposal, the Networked Music Performance mediated by co-creative agent provides the exchange of musical information in real-time and induces adaptative actions from both geographically distributed musicians and the artificial agent itself. In this way, we project such musical experience to bring new meanings to NMP experiences.

Next, we discuss the current technological tools in NMP research field.

3. CURRENT TECHNOLOGY IN NETWORKED MUSIC PERFORMANCE

The design and development of Networked Music Performance tools can be divided into three parts: a) the design and development of systems and/or interfaces; b) the design and development of protocols to establish communication between distributed agents over the network; c) the design of sound artifacts and the performance itself;

From the late 1990s through the early years of the 21st century, researchers focused on projects that aimed to achieve high-quality, real-time Networked Music Performance [14]. This performance, in addition to raising questions from a technological point of view, also led researchers to analyze the cultural implications that such installations imposed on musicians and producers [14], establishing strategies to increase the level of interactions between musicians who collaborate through network connections.

Thus, the challenges imposed at this stage of the NMP research field resulted in three approaches to designing NMP tools [14], namely:

- **Realistic Jam Approach:** When real-time live musical interactions are crucial, requiring geographically displaced musicians to come as close as possible to feeling like they are playing in the same space;
- **Latency Accepting Approach:** It considers the internet as a decentralized and space-independent medium. Therefore, efforts are focused on finding new forms of delayed musical interaction;
- **Remote Recording Approach:** It involves producing music using the internet as a medium for remote recording sessions, thus overcoming latencies for real human-to-human interactions to occur is not the goal.

As an example of technologies and tools included in each of these approaches we can mention: *Soundjack* [1] [15], *Ninjam* [2], and *VSTunnel*.

From this scenario, and with the increasingly accelerated revolutions in processing capacity and connection between devices, we have seen in recent decades the increasingly prominent development of tools and technologies aimed at distributed and web-based music systems for performance.

Moreover, the advancement of techniques and tools for the development of web platforms – made possible by the architectures, protocols, infrastructures and frameworks that are standardized in the market – makes us divert our focus from simply exploring and probing the possibilities and opportunities inherent in distributed and web-based music systems [3], allowing us to delve deeper into the characterization and implementation of experimental platforms that can function as a shared space for artists and researchers. In that sense, *Pteroptyx Malaccae* and *Clock(s)* [3] are good examples because they allow modularization as clients of a distributed system, reusing and adapting many existing software components such as clock synchronization, scheduling, stream processing or functionalities offered by existing frameworks.

However, although these technologies aim to simplify and democratize the experimental implementation of platforms dedicated to distributed and interactive web-based musical systems for artists and researchers in exploratory tasks, they were emergent technologies and had drawbacks related to the simplification of experimentation for non-specialist developer users such as composers, researchers or computer music designers.

Another issue currently imposed on the technological scenario in the NMP field is the creation of dynamic network infrastructures capable of connecting anything – whether they are physical or virtual objects – through different means of communication. This enables the collection of any type of information that can further be communicated and controlled via the Internet. The sensitivity and ability of devices to react and change their behavior based on the context and environment in which they are inserted, as well as the way they treat and exchange data with other devices that are connected to them, is a factor that it has been explored and placed in question.

As an example of this kind of technology is *Sunflower* [16, 17], an environment design that tries to contribute to solving the recurrent problems related to standardization, interoperability, privacy, and security between devices in a context of IoT musical technologies. Also, it handles different pieces of information, such as audio, video and control, in order to expand the capacity of non-musicians to participate and assist in an artistic creative process [17].

In order to support the interaction with all sort of musical things, Sunflower's management layer was inspired by other tools that work similarly, such as *Libmapper* [18] and *Medusa* [19], and also in tools present in musical desktops, like ALSA MIDI [3] and QJackCtl [4] [16]. Nevertheless, even though *Sunflower* is a playful and intuitive platform, which allows multiple users with different objectives and skill levels to use it without any major problems, it does not necessarily guarantee a good user experience in controlling the sound artifacts or other piece information in musical activities. This, however, is out of the scope of this paper.

Recently, given the social isolation caused by COVID-19 and motivated by the challenge of creating and interacting

[1] https://www.soundjack.eu/
[2] https://www.cockos.com/ninjam/
[3] https://www.alsa-project.org/
[4] https://qjackctl.sourceforge.io/

with other artists remotely, *grainBIRD* [20] was developed having as a concept a haze of mobile devices generating granular sounds via OSC messages within a network.

grainBIRD's network communication was based on IoT, Cloud computing and Fog computing paradigms, being derived from the concept of the laptop orchestra. The design of *grainBIRD*'s app applied the notion of cloud and fog computing architecture, in order to enable both local interactions and performances over a network of mobile devices [20].

As much as this technology interacts with the possibility of co-creation through the Internet and also with the paradigm of Ubiquitous Music, its multiple performance configurations can be further explored to obtain significant artistic results to promote engagement among performers connected to the network.

To summarize, it is clear that although the area of NMP has produced an extensive literature, its efforts are still related to technical aspects inherent to the development of the tools that are proposed. However, as musical practices offer a valuable mean for exploring the connections between perception, action, and meaning within the dynamic interplay among performers, audiences, and their environment [21], we see in this gap an important contribution to researches on perceptual aspects in the context of Networked Musical Performance.

It is in this context, thus, that the *JANIS* system is proposed, fostering the engagement and adaptation of performances in a NMP context based on an architecture which will be further explained in the next section.

4. ARCHITECTURE AND DEVELOPMENT STRUCTURES

In this section, we will focus on describing the architecture of *JANIS* (Joint Audio Networked Interactive System), a prototype system that aims to manage interactive performances through the web from an artificial co-creative agent.

Also, we will highlight the applied digital signal processing and the similarity calculation between pre-recorded signals and real-time signals to guarantee the artificial agent its independence for decision making.

Finally, we combined the results obtained from the digital signal processing, along with the concept of co-creativity explained in Section 2, to highlight some preliminary results.

4.1 User interaction and system's architecture

The starting point for using the *JANIS* system is the user interface created to perform interactions based on predefined actions. This interface, as shown in Figure 1, has buttons that, when clicked by the users, trigger actions such as recording, analyzing, comparing audio signals, among others.

These actions, in turn, are perceived by the users through a chat, in which they receive messages from the agent indicating which actions it has just performed, or through the signal graph, which indicates the capture of the audio and

Figure 1: System's user interface which makes interaction between user and co-creative artificial agent possible.

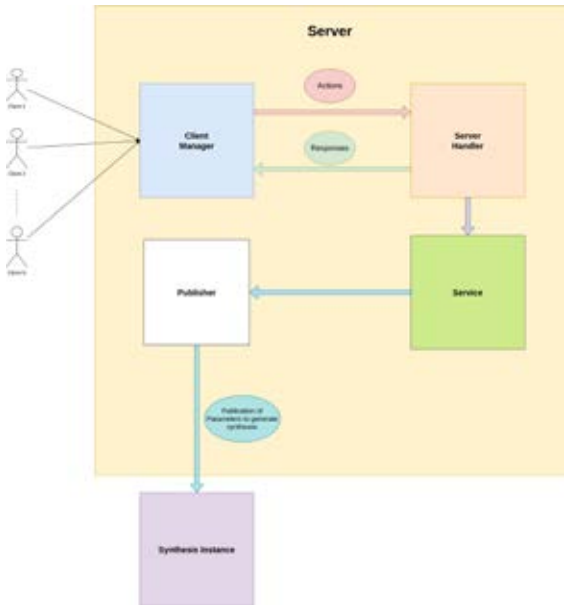


Figure 2: Architecture schema of co-creative system.

its reception by the agent. A more detailed description of these actions and the routines performed both by the users and the server side can be seen in the Table 1.

In order to establish communication between the client and the server, in this case between the user and the co-creative agent, the Websocket protocol was employed. This protocol selection was chosen because, unlike the HTTP protocol – which terminates its connections upon receiving a response – Websocket maintains two-way communication, ensuring continuous connectivity until termination is initiated by one of the parties [22].

Moreover, the decision to use the Websocket protocol stems from its ability to efficiently handle a considerable volume of messages within a defined time frame [22], consequently increasing the system's capacity to process the continuous flow of information that a performance can produce.

Table 1: Actions and routines make in front-end and back-end

Actions	Front-end routine	Back-end routine
record	Send audio data to back-end process it	Process the audio data and stores it in a wave file, returning a warning message to the user that audio has been recorded
analyse	Send a message to the back-end requesting analysis of recorded audio	Extract parameters from the signal and save it in a file, returning a message to the front that the audios have been analyzed
open	Send this message to start sending real-time audio signal to the back	Start monitoring audio data in real time
compare	Send this message to the back to start comparing the audio files with the captured signal in real time	Process captured audio data in real time and compares it with data pre-recorded by the record command
stop	Sends message to stop comparison	Stop comparing data captured in real time and pre-recorded audio
close	Send message to close real-time audio capture	Close real-time audio capture

By choosing this type of protocol, therefore, our focus was on ensuring means for external events to the system, in this case the musical performance, to determine the program's execution flow, as its different formats, in this case its different symbolic interactions, made the system listens to events that happen over time and react to them as they occurred.

This allowed us to structure an architecture with greater flexibility and responsiveness, in addition to seeking better performance and a consistent structure capable of supporting the volume of data to be processed.

In this way, the server implementation was defined based on four layers, namely:

- **Client Manager:** Responsible for managing user connections, as well as sending messages to a specific user, or to other connected users;
- **Server Handler:** Responsible for handling server routines and managing messages received from clients and send to be processed by service layer;
- **Service:** Responsible for carrying out actions requested by users;
- **Publisher:** Responsible for publishing messages or data to be executed or processed by third-party systems.

To complement the scope of the system, as shown in Figure 2, the server sends information to third-party systems to synthesize sounds using different processes. In our case, we choose Supercollider as an instance to produce FM, Granular, and Wavetable synthesis.

It is worth mentioning that this communication is carried out using the Open Sound Control (OSC) protocol, which makes it possible to use the system with other types of sound synthesis programs, such as Max/MSP or PureData.

This sending of data is carried out by calculating the similarity between the sound captured in real time and other pre-recorded sounds, which will be better highlighted in the next subsection.

4.2 Digital signal processing and computing similarity

As the structured architecture proposed and explained in subsection 4.1 aims at a reactive approach to the various symbolic interactions that occur between human and artificial agents throughout the performance, both DSP analysis and similarity calculations should follow this same principle, as they should not interrupt or block the data flow to be processed.

Thus, the starting point for performing calculations on both pre-recorded and real-time audio signals s is given by the root mean square of them, as shown in (1):

$$f(t) = \sqrt{\frac{1}{K} \cdot \sum_{k=t\cdot K}^{(t+1)\cdot K-1} s(k)^2}, \tag{1}$$

where K is the number of samples we have in a frame and t is the frame where root mean square (RMS) will be computed.

After that, and as shown in (2), the gradient is computed using second order accurate central differences in the interior points and either first order accurate one-sides (forward or backwards) differences at the boundaries:

$$\nabla f(t_i) = \frac{f\left(t_{i+1}\right) - f\left(t_{i-1}\right)}{2h} + \mathcal{O}\left(h^2\right), \tag{2}$$

where t_i is the frame, h is a non-homogeneous step size and $\mathcal{O}\left(h^2\right)$ represents the order of error proportional to h^2.

To avoid huge distortions, and to extract meaningful results, this gradient is normalized using (3):

$$X_{\text{norm}} = \frac{X - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}, \tag{3}$$

where X is the array resulted from gradient's calculation.

The data resulted in this gradient computation is analyzed in order to find the peaks of the signal, which takes into account a one-dimensional data series defining a local maximum as a point x such that:

$$x_{i-1} < x_i \text{ and } x_i > x_{i+1}. \tag{4}$$

To avoid inappropriate local maxima, we can optionally select conditions such as height, limit, distance, prominence and width that are adjusted according to processes and data obtained in each performance.

These values are then stored in files that can be used throughout the performance to calculate the similarity with the signals captured in real time.

To calculate the similarity between these signals, the cosine of the angle produced by their arrays is computed, as shown in (5):

$$\text{similarity}(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||}, \tag{5}$$

where $\vec{u}$ and $\vec{v}$ are, respectively, the vectors of input and pre-recorded signals peaks computation, $\cdot$ is the dot product and $||||$ is the Euclidean norm.

The closer the value of this angle's cosine is to zero, given a minimum threshold also defined from the data obtained in each performance, the more similarity is detected by the artificial agent.

This similarity, therefore, allows this artificial agent to make decisions throughout the performance and contribute co-creatively with the performers in the production of emergent processes.

Also, given the indeterminacy of each result obtained in the various performances, musicians have to adapt to the adverse conditions that each one may generate, even more so when thinking about a geographically distributed performance context.

In other words, this allows us to raise questions about what conditions musicians are able to adapt and to adjust their anticipation according to the confirmations or surprises that occur, the ability to enjoy artistic freedom, and the amount of interest inspired by the productions of the machines based on reciprocal intentionality.

4.3 Digital Signal Processing Performance and Discussion

We carried out some tests of JANIS system's digital signal processing calculation – to measure its feasibility in real-time performances. In this setting, a classic guitar standalone solo performance was evaluated.

Figure 3 shows the computation of the peaks obtained from the RMS gradient of the guitar signal, applied to different attack and sound production modes.

To detect the peaks, tests were applied with a threshold varying from 0.3 to 0.5 according to the variance that each attack mode could produce.

Furthermore, we decided to analyze the spectrogram of each signal, as shown in Figure 4, in order to identify which scenarios detection could be inefficient given the complexity of the sound spectrum formed.

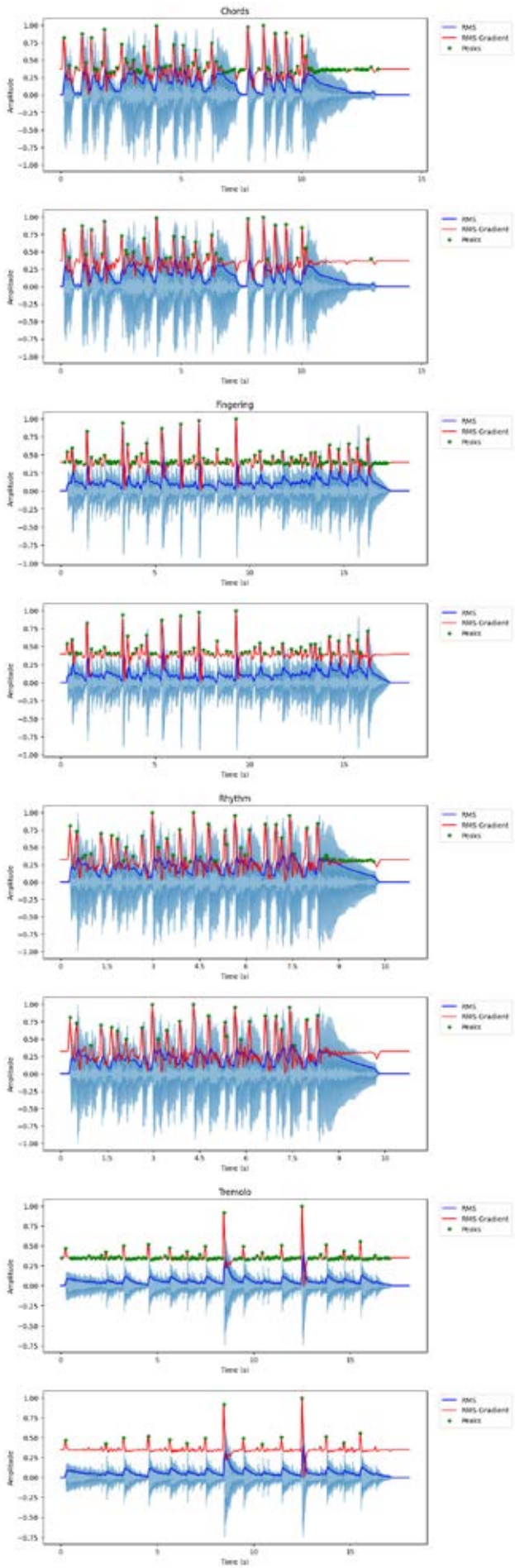


Figure 3: Peak finding results applied in different attack and sound production modes.

From the graphs presented, it is possible to evaluate that sounds that are more defined in terms of their spectrum, such as those presented by *fingering* and *tremolo*, have less inaccuracy in detecting their peaks, whether they are at points of greater magnitude or high variance.

For less defined sounds, such as *chords* and *rhythm*, depending on the defined threshold values, we see some inaccuracy in finding peak points in the RMS gradient.

This shows that, depending on the parameterization chosen for each performance, the actions of the artificial agent can vary significantly according to what it is "hearing". This means that the musician also needs to be attentive and present, being able to produce meaningful interactions based on the cognitive stimuli that the artificial agent may provides.

Also, by projecting the similarity of the input signal based on the greater or lesser clarity in the segmentation of prerecorded signals, we can control the synthesis parameters to generate distinct sounds from a given configuration. For example, if the input signal has high energy and the segmentation is clearer, we can work on the harmonicity, rhythm or even the amplitude of the output signal to be clearer, periodic or harmonic. If the signal has less energy and the segmentation is less clear, these same characteristics can result in sounds that are less harmonic and have a more dynamic rhythm. However, these characteristics can be explored based on the different artistic intentions that artists want to produce.

From the perspective of distributed sound performance, the synchrony and complexity of the resulting synthesis should not be the starting point, but rather the different configurations that allow agents to listen to and interact with each other.

The network latency is neither significant nor it is the cause of engagement, putting into play the synergy that must be managed by both the human and the artificial agent to produce novel and engaging musical expressions.

From the discussion here, it is possible to understand that *JANIS* has a great potential to allow multiple performance configurations that can produce co-creative process between both human and artificial agents over a NMP context and can also be part of several distributed performance configurations.

5. CONCLUSIONS

In this paper, we focused on discuss the possibilities to apply the concept of co-creativity and symbolic interactions in the context of Networked Music Performance in order to promote the involvement and engagement of geographically distributed musicians during the performance.

The use of artificial co-creative agents can be effective in promoting interaction between human agents, since their active interaction provides the creation of projection spaces that can be interesting for both to share their actions and efforts. This happens because both human and artificial agents are in a situation of adaptive interdependence and incorporating memories that promote a rich and dynamic creative environment for their interactions. This phenomenon has been widely explored in recent years in the context of human-computer co-creativity, and could be used to promote meaningful experiences in NMP contexts.

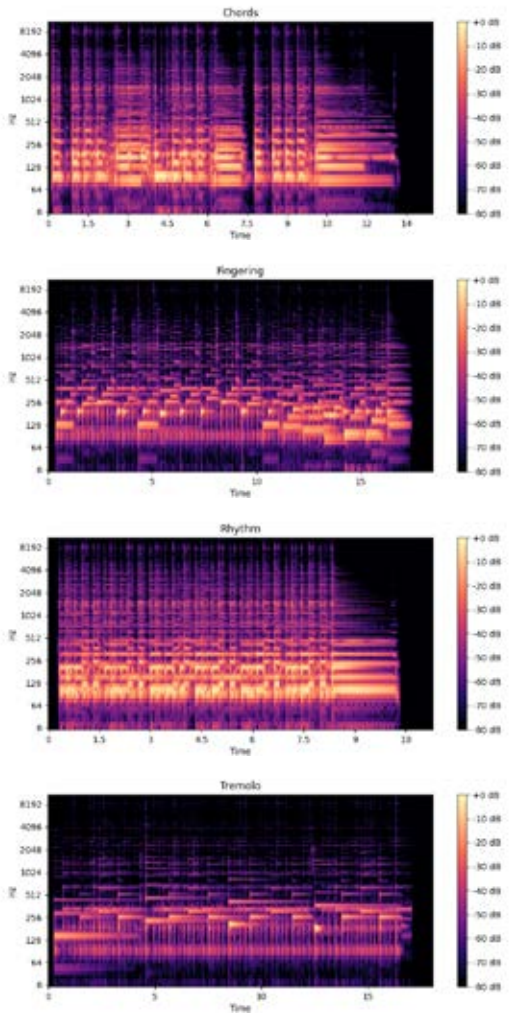


Figure 4: Spectrogram analysis of this guitar attack and sound production modes.

Furthermore, by proposing the prototype of the *JANIS* system, which, based on the tests carried out, has great potential to generate multiple performance configurations, our focus converges on studying cognitive and artistic strategies to promote engagement and adaptation of musicians in geographically distributed performances.

Finally, we intend to apply more tests to this system to understand its possibilities and limitations in addition to developing its communication with other digital musical instruments and its application within artistic research involving musicians and dancers, which motivated the development reported here.

Acknowledgments

This work is funded by the FCT - Foundation for Science and Technology, I.P./MCTES through national funds (PIDDAC), within the scope of CISUC R&D Unit - UIDB/00326/2020 or project code UIDP/00326/2020*

Also, this work is part of the funding base project activities of the Center for Technology and Innovation (CTI)

Pedro Nunes Institute (IPN), according to the Call for Proposals (AAC) No. 03/C05-i02/2022, within the scope of the Portuguese Recovery and Resilience Plan (PRR).

6. REFERENCES

[1] A. S. D. Almeida, R. Vieira, R. S. Oliveira, and F. L. Schiavoni, "Desafios da Internet das Coisas Musicais," in *Anais do XVIII Simpósio Brasileiro de Computação Musical (SBCM 2021)*. Brasil: Sociedade Brasileira de Computação - SBC, Oct. 2021, pp. 249–252.

[2] K. Nishida, A. Yuguchi, K. Jo, P. Modler, and M. Noisternig, "Border: A Live Performance Based on Web AR and a Gesture-Controlled Virtual Instrument," Jun. 2019, publisher: Zenodo. [Online]. Available: https://zenodo.org/record/3672860

[3] B. Matuszewski and F. Bevilacqua, "Toward a Web of Audio Things," in *Sound and Music Computing Conference*, Limassol, Cyprus, Jul. 2018.

[4] L. Turchet, C. Fischione, G. Essl, D. Keller, and M. Barthet, "Internet of Musical Things: Vision and Challenges," *IEEE Access*, vol. 6, pp. 61 994–62 017, 2018, conference Name: IEEE Access.

[5] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An Overview on Networked Music Performance Technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016, conference Name: IEEE Access.

[6] G. Assayag, "Human-Machine Co-Creativity," in *in Bernard Lubat, Gérard Assayag, Marc Chemillier. Artisticiel / Cyber-Improvisations. Phonofaune, 2021, Dialogiques d'Uzeste.*, May 2021.

[7] O. Harold, "The Concept Of Creativity in Art," *The British Journal of Aesthetics*, vol. 19, no. 3, pp. 224–231, 07 1979. [Online]. Available: https://doi.org/10.1093/bjaesthetics/19.3.224

[8] M. A. Boden, Ed., *Dimensions of Creativity*. The MIT Press, Jun. 1994.

[9] R. K. Sawyer and D. Henriksen, *Explaining Creativity: The Science of Human Innovation*. Oxford University Press, Jan. 2024.

[10] M. A. Boden, *The Creative Mind*, 2nd ed. 11 New Fetter Lane, London EC4P 4EE: Routledge, Feb. 2004.

[11] A. Williams, "Behind the Idea: Meta-Models for Creativity," in *Intersecting Art and Technology in Practice: Techne/Technique/Technology*. Taylor & Francis, Dec. 2016, pp. 182–197.

[12] G. Assayag, "Sex Machine: IA CoCreative et position du sujet," Jan. 2024. [Online]. Available: https://www.youtube.com/watch?v=LWT99riOWeQ

[13] M. T. Llano, M. d'Inverno, M. Yee-King, J. McCormack, A. Ilsar, A. Pease, and S. Colton, "Explainable Computational Creativity," 2022.

[14] A. Carôt, P. Rebelo, and A. Renaud, "Networked music performance: State of the art," in *Audio Engineering Society Conference: 30th International Conference: Intelligent Audio Environments*, Mar 2007.

[15] A. Carôt, C. Hoene, H. Busse, and C. Kuhr, "Results of the fast-music project—five contributions to the domain of distributed music," *IEEE Access*, vol. 8, pp. 47 925–47 951, 2020.

[16] R. Vieira and F. L. Schiavoni, "Managing an IoMusT environment and devices," in *11th Workshop on Ubiquitous Music (UbiMus 2021)*, ser. Proceedings of the 11th Workshop on Ubiquitous Music (UbiMus 2021). Matosinhos, Portugal: g-ubimus, Sep. 2021, pp. 32–44.

[17] R. Vieira, D. C. Muchaluat-Saade, and F. L. Schiavoni, "Sunflower: An Interactive Artistic Environment based on IoMusT Concepts," in *Proceedings of the 2022 ACM International Conference on Interactive Media Experiences*, ser. IMX '22. New York, NY, USA: Association for Computing Machinery, Jun. 2022, pp. 245–248. [Online]. Available: https://doi.org/10.1145/3505284.3532818

[18] J. Malloch, J. Garcia, M. M. Wanderley, W. E. Mackay, M. Beaudouin-Lafon, and S. Huot, "A Design Workbench for Interactive Music Systems," in *New Directions in Music and Human-Computer Interaction*, ser. Springer Series on Cultural Computing, S. Holland, T. Mudd, K. Wilkie-McKenna, A. McPherson, and M. M. Wanderley, Eds. Cham: Springer International Publishing, 2019, pp. 23–40. [Online]. Available: https://doi.org/10.1007/978-3-319-92069-6_2

[19] F. L. Schiavoni and M. G. d. Queiroz, "Medusa: um ambiente musical distribuído," Master's thesis, Universidade de São Paulo, 2013. [Online]. Available: http://www.teses.usp.br/teses/disponiveis/45/45134/tde-08092014-172703

[20] J. Manzolli and E. H. Constantino, "grainBirds: Evolutionary granular synthesisers distributed in Fog Computing," in *Anais do XVIII Simpósio Brasileiro de Computação Musical (SBCM 2021)*. Brasil: Sociedade Brasileira de Computação - SBC, Oct. 2021, pp. 95–102.

[21] A. Linson and E. F. Clarke, "Distributed cognition, ecological theory, and group improvisation," in *Distributed Creativity: Collaboration and Improvisation in Contemporary Music*. Oxford University Press, Dec. 2017, vol. 2, pp. 52–69.

[22] G. Gonçalves, P. Bastos, and D. Oliveira, "O uso de websockets no desenvolvimento de sistemas baseados em uma arquitetura front-end com api," in *Anais da I Escola Regional de Sistemas de Informação do Rio de Janeiro*. Porto Alegre, RS, Brasil: SBC, 2014, pp. 57–63.

PERFORMING VOICES IN IMMERSIVE THEATER: KINESTHESIS 2.0 FOR EMBODIED PERFORMANCE

Tilemachos MOUSSAS (Τηλέμαχος Μούσας)[1], **Natalia KOTSANI** (Ναταλία Κωτσάνη)[2], and **Anastasia GEORGAKI** (Αναστασία Γεωργάκη)[1]

[1]**National and Kapodistrian University of Athens**, Athens, Greece
[2]**National and Technical University of Athens**, Athens, Greece

ABSTRACT

Kinesthesis 2.0 is an innovative interactive tool designed for augmenting the voice of singers and actors, enabling them to dynamically modify their voice in real-time during music theatre performances. This tool enhances immersive theatre by integrating technology and artistic expression, allowing for an interplay between verbal and gestural elements. Kinesthesis 2.0 focuses on achieving embodiment, where actors use multi-sensory stimuli, integrating voice, facial expressions, and gestures to create a comprehensive portrayal of characters. This embodiment is facilitated by the tool's ability to capture and translate these modulations, offering a dynamic interaction between auditory and visual storytelling.

Technically, Kinesthesis 2.0 uses Python, Open Sound Control protocol, and Max/MSP to map the facial movements into predefined gestures. MediaPipe is used for face and hand detection, while OpenCV processes the webcam feed, capturing head movements, facial expressions, and hand gestures. Through the gesture and facial expressions detector, the artist controls various effects (pitch shifter, multi-channel vocoder, reverb, multi-tap delay and reverse delay). Artistically, Kinesthesis 2.0 enriches the expressive capabilities of performers by amplifying and augmenting their voice, thus enhancing their ability to convey emotions and narratives. As a case study, the tool was used by the singers and actors in the theatre performance "Rayman Scream".

1. INTRODUCTION

Kinesthesis 2.0 combines a facial expressions and a hand gesture detection system developed in Python using Mediapipe with a four-voice vocoder, a pitch-shifter, a reverb, a multi-tap delay and a reverse delay developed using MAX/MSP. By tracking movements and using them to control the parameters of the vocoder and delay effects, a performer can create expressive vocal effects combined with a digital choir, by moving face, head or hands in specific ways, creating complex vocal timbres and rhythms. Kinesthesis 2.0 was tested and used during the rehearsals from the composer, the actors and singers of the theatre play "Rayman Scream", since December 2023.

By capturing bodily movements, Kinesthesis 2.0 facilitates a form of embodiment, wherein the performer's bodily gestures and facial expressions directly influence the auditory output, using different sensory modalities. This integration of gesture, technology, and artistic expression aims to immerse the audience in a multisensory performance, where the interplay between physical movement and vocalization enriches the overall narrative experience.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED RESEARCH

Gesture is a part of human communication, and it has been used as a form of non-verbal communication. It is a fundamental component of language that contributes meaningful and unique information to a spoken message and reflects the speaker's underlying knowledge and experiences. Theoretical perspectives of speech and gesture propose that they share a common conceptual origin and have a tightly integrated relationship, overlapping in time, meaning, and function to enrich the communicative context [1].

Gesture as an expressive tool has been a fundamental aspect of theatricality throughout history and constitutes a codified theatrical language performed not simply through speech [2]. Gestuality, is an extension of natural human behaviors, with actors using their bodies to convey meaning and emotion to their audiences. Gestures can be used to communicate a wide range of emotions and ideas, from joy and excitement to fear and sorrow, and they can be used to convey complex concepts and ideas that may be difficult to express through words alone.

For example on the southern coast of India, the Kathakali ritual theatre hand gestures called Mudras [3], are central to the performance. Divided into two primary components, Cholliyattam and Elakiyattam, the narrative unfolds through intricate gestures, body movements, and facial expressions. The significance of eyes in expressing emotions (Oculesics in Sathika Abhinaya) is emphasized, with early morning eye exercises shaping the actor's ability to convey emotional states. In Cholliyattam, enacted verses are brought to life through a sophisticated blend of Kinesics, Gait, Postures, and Gestures, complemented by the rhythmic repetition of sung verses. Elakiyattam, the silent pinnacle of Kathakali, showcases the actor's creative freedom to communicate without singing, utilizing a combination of non-verbal elements such as Kinesics, Gait,

Postures, Gestures, and Oculesics, along with characteristic grunts. These Mudras, with their precise and elaborate movements, convey profound meanings, emotions, and ideas.

Actors are using a series of movements and poses to convey meaning and emotion to their audiences. While some gestures are traditional schemata, there are also "less obvious" gestures that are just as important in conveying meaning on stage. These everyday gestures can be found at the heart of a scene, revealing an actor's character and emotions in subtle ways. In a study devoted to physical contact in tragedy, Maarit Kaimio [4] explores these less obvious gestures, including the gesture of "nursing" that appears in a scene from Euripides' Orestes. This gesture, which may seem insignificant at first glance, reveals much about the characters and their relationships.

In the 20th century, the use of gesture in theater underwent another significant evolution, as Jerzy Grotowski, Eugenio Barba, and Peter Brook explored new ways of using gesture to create powerful and immersive theatrical experiences. Grotowski, developed a technique known as "poor theater," highlighting the use of the body and voice to create intense and emotionally charged performances. One of the most interesting developments in recent years has been the use of motion-capture technology in theater, which allows actors to create complex and realistic movements that can be projected onto a screen or other digital medium [5].

In achieving expressive electroacoustic music, a deep understanding of the morphological relationships between physical gestures and their corresponding sonic outcomes is necessary. When a morphological relationship is deterministic, it means that there is a consistent and logical correspondence between the performer's intentions, expressive gestures, and the resulting sound. For example, a gentle short gesture of our fingers could produce a subtle and intimate sound, whereas a violent movement could create a more dramatic and intense sound. Research has shown that audiences heavily rely on physical gestures to understand the expressive intent of a performance [6] and that electronic music performances ought to demonstrate a coherent correlation between the magnitude of a control gesture and the ensuing acoustic outcome. [7].

3. MULTIMODAL INPUTS AND AUDIENCE PERCEPTION

In immersive theatre, "immersive" not only means "a state of being immersed", but also "providing information or stimulation for a number of senses". There are many types of sensory stimulation, including visual, aural, and tactile [8]. Kinesthesis 2.0 creates an immersive and multisensory experience for both performers and audiences. Research in multi-sensory processing and cross-sensory cognitive modes has shown how the brain constructs perceptions by combining congruent inputs and segregating incongruent ones. These studies have explored terms like "synesthetic association" or "cross-modal correspondence" to describe how different sensory dimensions correspond to each other [9].

For example, synesthetic correspondence involves non-redundant sensory dimensions, such as the association between sound frequency and brightness. By contrast, cross-modal correspondences include both synesthetic associations and those between redundant stimuli perceived through different sensory modalities, such as the auditory and visual length of an event [10]. Spectators are not merely listeners, they become observers of a synchronized dance where gestures become audible, creating an immersive encounter. Music perception is achieved through interaction with other sensory fields, as a result, the audience can activate memories, images and feelings [11]. An immersive experience can manifest in different states: "engagement, engrossment, and total immersion," each representing a gradient of immersive experiences. Interactivity and immersion are closely related in performance, interactivity can enhance immersive experiences by fostering deeper audience involvement [12, 13]. By applying sonic interaction design, this tool aims to explore the methods for engaging audiences with immersive experiences by extending the interactivity to the auditory environment.

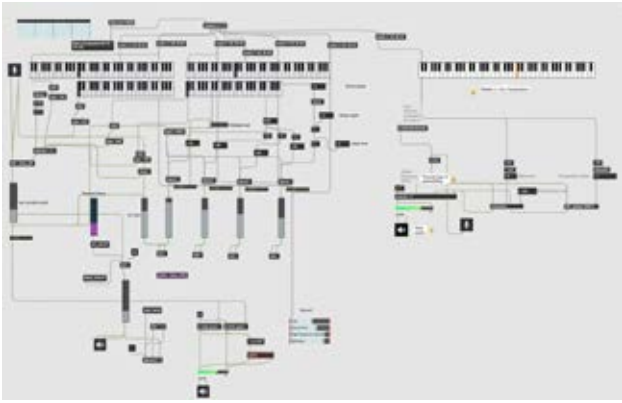
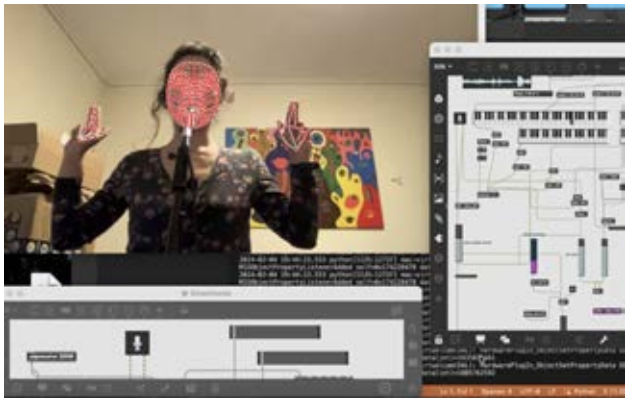


Figure 1. Kinesthesis 2.0 Max/MSP Patch

4. ARCHITECTURE OF THE SYSTEM KINESTHESIS 2.0

The Kinesthesis 2.0 simple interface, through the Max/MSP patch, offers various pitch-shifting options, including transposer, gizmo or bypass. Additionally, the user can choose a specific type of a MIDI instrument to track the voice, choosing between "harmonic", "in-between" or "percussive" mode. In the Kinesthesis 2.0 interface, the user can control through various parameters the vocal effects in real-time. This includes a four-voice harmony through four individual vocoders, assigning each vocoder to a specific voice within the harmony and modulating their parameters accordingly. The interface offers also various preset options for reverse delay customization (for the reproduction of recorded phrases in reverse playback), a multi tap delay and a parametric reverb effect.

The real-time control of the sound parameters is achieved by using hand gestures and facial expressions. The OSC (Open Sound Control) messages for facial movements, such as head tilts, eye movements, and mouth gestures (mapped to predefined gestures such as 'One' 'Two'



A Mezzo Soprano using Kinesthesis 2.0

'Three' 'Four' and 'Five') are sent to an IP address and port. The gestures are continuously analyzed and when a match is found, the corresponding OSC message is dispatched. The Max/MSP patch receives OSC messages transmitted over a network using the User Datagram Protocol (UDP). Subsequently, the received list of data is broken down into individual messages, with each message corresponding to one of the parameters intended for processing. This allows us to dynamically control various aspects, such as the pitch of each vocoder, the percentage of delay effect (specifically, multi-tap and reverse delay), as well as the size, decay time, high-frequency damping and diffusion of the reverb.

The Python code and the MAX/MSP patches of Kinesthesis 2.0 can be found in a public repository [1].

5. AESTHETIC ISSUES AND CONCLUSIONS

Kinesthesis 2.0 was used for the first time in the theatre performance "Reyman Scream" (March 2024, Theatre "Sfendoni", Athens), as a vocal augmentation system, offering to the performer the ability to improvise and composing music in real-time. On the rehearsals, the theatrical ensemble has used Kinesthesis 2.0 real-time voice automation and expressive features to dynamically modulate the soprano singer's and the main actor's vocal output, creating a harmonious blend and simulating a digital choir, synchronizing vocal elements with dramatic expressions. This integration demonstrated Kinesthesis 2.0 functionality and enriched the collaborative dynamics in the context of a theatre performance. A demo from the theatre play rehearsals is uploaded in a public repository [2].

One of the most challenging aspects of the Kinesthesis 2.0 interface is the use of real-time vocal input. While audiences generally respond positively to these moments, there are some issues that need to be addressed in order to optimize the use of this feature. For example, audiences may have a preconceived notion that "singing" will occur when the performer is using the microphone. This can lead to disappointment when the performer focuses on gesture control without live vocal input. To address this issue, performers can be trained to use vocal input more effectively and to communicate with the audience about what they are doing.

Another issue with real-time vocal input is that it can become difficult for audiences to understand what the vocalist is doing when the composition becomes dense. This is because the voice is traditionally perceived as a solo instrument, which should be heard above a background texture. The microphone is a key component of the Kinesthesis 2.0 interface, but it can also be a source of confusion for audiences. The microphone directs focus towards the spatial positioning of the voice and the organization of surrounding auditory elements. Nevertheless, within the Kinesthesis 2.0 interface, the function of the live voice deviates from established contextual models. In this framework, the vocal element can undergo substantial transformations, resulting in dense textural accompaniments, and the degree of processing may reach an intensity wherein the original voice becomes challenging to discern. To address this issue, composers can create contrasting sections of the composition where the texture becomes more minimal and the connections between gesture and sonic outcome are more transparent.

Furthermore, it is important to consider the potential ethical implications of using the Kinesthesis 2.0 interface in a performance context. For example, if the Kinesthesis 2.0 interface is used to process and manipulate the performer's voice in real-time, it could be argued that this violates the performer's right to control their own artistic expression. It is important to address these concerns and to ensure that the use of the Kinesthesis 2.0 interface is transparent and respectful of the performer's autonomy.

Music perception is achieved through interaction with other sensory fields. Musical meaning might be constructed largely by synesthetic processes, where the sensory associations from sound activate memories, images, and emotions [9]. As feedback is gathered and new technologies and techniques are developed, the interface will continue to evolve and improve. By embracing a spirit of experimentation and continuous improvement, the Kinesthesis 2.0 interface has the potential to revolutionize the way that vocal performances are created and experienced.

Figure 2. Performing with Kinesthesis 2.0

[1] https://github.com/Tilemachos88/Kinesthesis-2.0
[2] https://drive.google.com/file/d/1lMhyEjLZOvChCnfAloLseboFkz4Sjzu9/view?usp=sharing

6. REFERENCES

[1] S. Clough and M. C. Duff, "The role of gesture in communication and cognition: Implications for understanding and treating neurogenic communication disorders," *Frontiers in Human Neuroscience*, vol. 14, p. 323, 2020.

[2] H. Roisman, *The encyclopedia of Greek tragedy*, 2014.

[3] E. Barba and S. Sanzenbach, "The kathakali theatre," *Tulane Drama Review*, vol. 11, no. 4, pp. 37–50, 1967.

[4] M. Kaimio, "Comic violence in aristophanes," *Arctos–Acta Philologica Fennica*, vol. 24, pp. 47–72, 1990.

[5] J. Grotowski *et al.*, "Towards a poor theatre, ed," *Eugenio Barba (Copenhagen: Odin Teatrets Forlag*, 1968.

[6] G. Emerson and H. Egermann, "Gesture-sound causality from the audience's perspective: Investigating the aesthetic experience of performances with digital musical instruments." *Psychology of Aesthetics, Creativity, and the Arts*, vol. 12, no. 1, p. 96, 2018.

[7] D. Wessel and M. Wright, "Problems and prospects for intimate musical control of computers," *Computer music journal*, vol. 26, no. 3, pp. 11–22, 2002.

[8] J. Machon, *Immersive theatres: Intimacy and immediacy in contemporary performance*. Bloomsbury Publishing, 2017.

[9] G. F. F. Bragança, J. G. M. Fonseca, and P. Caramelli, "Synesthesia and music perception," *Dementia & neuropsychologia*, vol. 9, pp. 16–23, 2015.

[10] N. Bien, S. Ten Oever, R. Goebel, and A. T. Sack, "The sound of size: crossmodal binding in pitch-size synesthesia: a combined tms, eeg and psychophysics study," *NeuroImage*, vol. 59, no. 1, pp. 663–672, 2012.

[11] J. S. Snyder, C. M. Schwiedrzik, A. D. Vitela, and L. Melloni, "How previous experience shapes perception in different sensory modalities," *Frontiers in human neuroscience*, vol. 9, p. 594, 2015.

[12] E. Brown and P. Cairns, "A grounded investigation of game immersion," in *CHI'04 extended abstracts on Human factors in computing systems*, 2004, pp. 1297–1300.

[13] N. Mount, C. Chambers, D. Weaver, and G. Priestnall, "Learner immersion engagement in the 3d virtual world: principles emerging from the delve project," *Innovation in Teaching and Learning in Information and Computer Sciences*, vol. 8, no. 3, pp. 40–55, 2009.

REAL-TIME PIANO ACCOMPANIMENT MODEL TRAINED ON AND EVALUATED ACCORDING TO HUMAN ENSEMBLE CHARACTERISTICS

Kit ARMSTRONG[1], **Tzu-Ching HUNG**[1], **Ji-Xuan HUANG**[1], and **Yi-Wen LIU**[1]

[1]**Acoustics and Hearing Group, Dept. Electrical Engineering, National Tsing Hua University**, Hsinchu, Taiwan

ABSTRACT

With a view towards the goal of modelling the performance of classical music by human musicians, we have set our focus on the question of collaborative music-making in a MIDI environment. In previous work, we have presented a model that plays a part of a score in real time together with a live musician playing another part. We trained it to resemble human musicians faced with the same task, by tuning its systems built around a set of Kuramoto oscillators. Here we chose 3 musical works and conducted experiments collaborating with a variety of pianists and record the resulting performances as well as the testers' subjective impressions. We reconciled each performance with the corresponding music score, thereby defining a dataset which we call an "interpretation". In addition to subjective evaluation, we introduced objective criteria in the form of discriminants that classify interpretations as being the result of human-human interaction or of human-machine interaction. We considered the following qualities: desynchronization, jerkiness, and velocity curves. Our trained model performed similarly to humans with respect to the first two discriminants, but significantly differently with respect to the last. In light of this, it is notable that our experiment subjects often failed to correctly distinguish the two classes.

1. INTRODUCTION

Musical automata have exerted a fascination on musicians and music lovers for centuries. In addition to their use for recording and reproducing music, the possibilities of collaborative performance involving humans and machines together have received increased attention in recent years. Creating interactive capabilities in accompaniment systems has become a lively area of research [1].

An obvious first issue to be tackled concerns enabling a machine to recognize and process live human performance, in order to be able to interact musically with it. An important part of this is perceiving and "understanding" rhythm, for which various approaches to machine listening have proven fruitful, such as conducting post-factum probabilistic analyses of perceived onsets [2].

Our direction of focus is guided by the goal of creating an "AI musician", thus necessitating processing in real time. In the case of Western classical music, the nature of the task is shaped by the existence of a score, with which the performance is to be reconciled in a process called alignment or score-following. Effective methods have been devised for usage with audio performances [3] [4] or MIDI input [5]. Some primarily focus on the synchronization of timing, while others additionally capture various features of expression from human performances [6]. The development of synchronization systems has often been inspired by biological phenomena with little apparent connection to music [7]. In particular, the Kuramoto model [8] has been adapted and extended to describe observed aspects of collaborative music making [9].

The complementary part of real-time musical collaboration involves creating an output based on the information received from the human performer and on the indications of the score. Approaches range from modifying, or "time-warping" an existing audio recording to fit with the instantaneous timing of the particular performance [4], to generating a new digital performance of the accompaniment part from the score. Particularly germane to our present approach is the work of Cancino-Chacón et al. [6], which envisages a musician playing a physical piano that automatically accompanies him/her, responding to spontaneous changes in tempo, dynamics, and expression. In pursuing a similar goal, we have started from a more modular approach, by focusing first on the rhythmic aspect of collaborative music making.

In our previous work [10], we developed and trained a model on data that we collected from human musicians whom we asked to perform, in a controlled environment, a task identical to the intended application of the model. Our work thus does not take perfect synchronization as the ideal, but rather endeavors to imitate human responses. In this paper, we describe a set of experiments that were designed for testing the model's performance in playing along with human pianists. The rest of this paper is organized as follows: Sec. 2 defines our data structure for representing musical interpretation in the MIDI environment; Sec. 3 reviews the construction of the present model, and how it incorporates the Kuramoto coupling equations; Sec. 4 describes our experiments; Sec. 5 presents objective evaluations of their results; our experiment subjects' impressions are reported in Sec. 6; and discussion and conclusion are given in Sec. 7 and 8, respectively.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

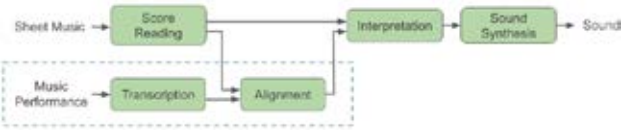


Figure 1. The role of the classical musician. The tasks within the dotted box apply when collaborating in an ensemble.

2. SCOPE

In this work, we represent scores as an ordered list of events, each following the format (type, note number, score position), where:

- "type" is "note-on" or "note-off";
- "note number" is the pitch designation according to MIDI;
- "score position" is the number of beats since the beginning of the piece.

The order of the list is determined by score position first, type second, and note number last. This method allows us to associate each relevant element of a score to a unique index number. In particular, each note is composed of a note-on event and a note-off event.

With our assumptions of a MIDI piano (excluding pedal) as our instrument, and error handling being outside of our scope, we can define an "interpretation" as the data of the score, as set out above, combined with information on how each event is performed. Specifically, to each event we associate the attributes of time and velocity. We may thus record an interpretation as an array of dimension $2N \times 6$, where N denotes the number of notes in the score, such that each row of this array follows the format of (index, type, note number, score position, time, velocity). This method also allows us to consider partial interpretations, where all score events are included, but not all events' attributes are defined.

In this framework, we may visualize the task of collaborative music making according to Figure 1.

3. MODEL

In this section we briefly recapitulate the essential aspects of our model. Its design is inspired by the application of Kuramoto oscillators [8] to biological phenomena [11], in particular to the synchronization of human tapping [12]. Although music obviously involves many complexities that do not appear in the referenced tapping task, we can make use of the same fundamental concept when we consider rhythm to be based on a more-or-less consistent beat, whose changes are subject to synchronization across players. We applied the model in the following environment: a human plays a one-voice part on a MIDI keyboard, while the model plays a second one-voice part, which we call here the "accompaniment" (though it may, in musical terms, instead represent the melody).

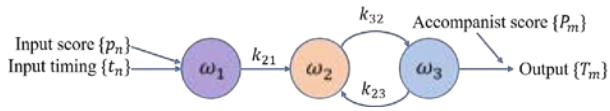


Figure 2. 3-oscillator Kuramoto model, as adapted for use in our model.

The model consists of 3 oscillators $\omega_1, \omega_2, \omega_3$ that are coupled as in Figure 2. Their positions are determined by the coupling equations:

$$\frac{d\theta_i(t)}{dt} = \sum_{j \neq i} k_{ij} \sin\big(\theta_j(t) - \theta_i(t)\big) + \Omega_i(t), \qquad (1)$$

for $i, j \in \{1, 2, 3\}$, where θ_i represent the positions of the respective oscillators, $\Omega_i(t)$ their intrinsic speed, and k_{ij} the coupling coefficients (with only k_{21}, k_{23}, k_{32} being non-zero, as per Figure 2).

For our application, since we treat music as consisting of discrete events, and no information is produced or exchanged between events, we must undertake certain non-trivial adaptations to the Kuramoto model. (See our previous publication [10] for details.)

By extrapolating the movement of ω_3, the Kuramoto oscillator system produces a series of predictions $\{T_m\}$ for the timings of subsequent events. Human instinct for the behavior that we are seeking to model served as the blueprint for integrating this system into our model. In particular, we presume that a human musician does not wait for each note before reacting; instead, he/she predicts how their partner will play, and how in consequence they should play, adjusting their prediction with every piece of new information that reaches their ears. We model this behavior by the following protocol:

1. the model makes a prediction of the timings of all future events up to a certain number of beats ahead;
2. the expected output based on this prediction is entered into a queue;
3. any time new information is received, a new prediction is made, and the following actions are executed:
   (a) output events that have already occurred will not be repeated, even if according to the new prediction they should be yet to occur;
   (b) output events that have not yet occurred, but that according to the new prediction should have already occurred, are carried out at once;
   (c) all other output events are updated in the queue according to the new prediction.

In this framework, it seemed reasonable to introduce a parameter t_r reflecting reaction time. That is, the above protocol is carried out after t_r has elapsed from the note having received. We denote by $\{T_m^*\}$ the timings of the output events hereby generated.

After performing a behavior-capturing experiment [10] with 20 subjects, for a total of 120 recordings, we searched for the parameters k_{21}, k_{32}, k_{23} and t_r that best fit the resulting data. Applying a gradient descent method over k_{ij}, t_r revealed a landscape with many local minima of similar value in the region of $5 \leq k_{ij} \leq 10$ and $t_r \approx 0.1$ seconds. We thus chose the values used in subsequent experiments to be $k_{21} = k_{32} = k_{23} = 5$, $t_r = 0.1$ seconds.

We programmed our model in Python, connecting it to MIDI input and output with RtMidi [1] . The program makes use of a clock (`current_time`), two queues (input and output), and 3 threads:

- `inputreading` records MIDI input events and their timing into the input queue;
- `calculating` matches each object as it appears in the input queue to the score, generating $\{t_n\}$, and computes θ_1 up until `current_time` by linear interpolation of the input onsets $\{t_n\}$. It then calculates θ_2, θ_3 according to Equation (1) and by extrapolating θ_3 produces a sequence $\{T_m\}$ of output timings;
- `worker` applies the protocol described above to output the notes at time $\{T_m^*\}$.

While error-handling is outside the scope of this research, we included a simple method to redeem minor mistakes. The `calculating` thread continuously keeps track of the last input note received, say, the nth note. When it receives (from `inputreading`) the next event, it first checks if the pitch corresponds to $(n+1)^{\text{th}}$ or $(n+2)^{\text{th}}$ input note according to the score. In the former case, the action proceeds normally; in the latter case, the $(n+1)^{\text{th}}$ note is inserted with an identical timestamp. In any other case, the event is simply ignored. With this method, errors of the following nature — replacing a note with a wrong note, playing an extra note, or omitting a note — will only cause at most a local disturbance.

Furthermore, we implemented a simple method for following the human player's dynamics: with each new input, the program calculates the average of velocities of all (correctly played) input notes whose score position falls within the last beat at the time of calculation. This produces a running average v_n, to which we apply a linear transformation $V_n = \alpha v_n + \beta$ (the parameters determined by perceived musical preference – when the output represents a melody, we set α to be greater than when it is an accompaniment). All subsequent output notes are then provided with velocity V_n. Upon the arrival of the $(n+1)^{\text{th}}$ note, the output velocity will become V_{n+1}, and so on.

4. EXPERIMENTS

To test the performance of our model, we designed a test in which a collaboration between a human and our model could be compared in near-identical conditions to a collaboration between two humans. We set up two MIDI keyboards, and connected them to a computer which ran on macOS Catalina (version 10.5.7) with Intel Core i5 1.6GHz Dual-Core CPU and 4 GB RAM, as in Figure 3. All sound was routed through the computer's MIDI output, connected to a set of speakers. This setup allowed for the sounds to be indistinguishable, whether they are occasioned by an input on either of the keyboards or generated by the computer.

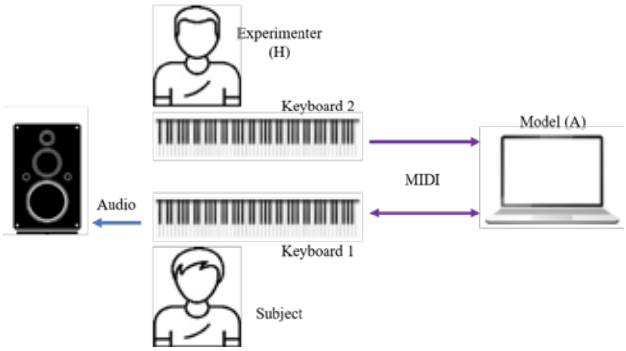


Figure 3. Hardware and human-interaction setup in the experiments

The experiment runs as follows: the subject is seated at keyboard 1 (KAWAI K-300 upright piano with AK-01 Touch Sensor), while an experimenter is seated at keyboard 2 (M-Audio ProKeys Sono 61), unseen by the subject. The subject is asked to play one part of a piece of music. The experimenter, without announcing their intention, chooses for the single trial in question either procedure A, in which the computer "plays" the second part by running the model, or procedure H, in which the experimenter plays it. [2] The subject, who has no knowledge of which case applies, is then asked whether he believes to have been playing with a computer or with a human. Listeners present in the room, but not looking at the operation of the experiment, are also asked whether they heard a human-computer or a human-human performance.

When the experimenter chose A, the program generated records of the following sequences:

$\{s_n\}$	the timing of the subject playing the input notes,
$\{t_n\}$	the input timing registered by the program,
$\{T_m^*\}$	the model-computed timing of output notes, and
$\{U_m\}$	the timing of the output actually being sent to the MIDI device,

and saved these in a series of `.txt` files. For H, another program was run, which simply recorded the inputs of both keyboards while transferring them for output through the speakers. For convenience of notation, we reuse the symbols $\{s_n\}, \{U_m\}$ for the timings of the subject and the experimenter, respectively.

For the above experiment, we selected 3 pieces of music:

1. W. A. Mozart: "Twinkle, Twinkle, Little Star" K.265, Theme

[1] https://spotlightkid.github.io/python-rtmidi/

[2] In order to avoid giving clues through external phenomena such as keyboard noise, in both cases the experimenter plays, but in the case A, his keyboard is not connected.

2. W. A. Mozart: “Twinkle, Twinkle, Little Star” K.265, Variation II

3. J. S. Bach/C. Gounod: “Ave Maria, Méditation sur le Prélude de Bach”

In (1) and (2), the participant was to play the right hand part while the computer (or experimenter) played the left hand. Each participant was asked to provide four extra notes at the beginning of each trial to indicate the intended tempo. In (3), the participant was to play the Prelude by Bach while the computer (or experimenter) played the melody by Gounod. To facilitate the task for the participants and minimize the confounding factor of their pianistic skill, for (1) and (2) they played a simplified version of the melody with only quarter notes.

We conducted the experiment with 12 participants, for a total of 80 trials. Each participant, with the exception of one, played each piece at least twice, and at least once with each partner (unknowingly).

5. DATA PROCESSING AND ANALYSIS

With the computer having recorded each trial, regardless of whether A or H was chosen, we proceeded to align the performances to the corresponding scores and produce "interpretations" in the form presented in Sec. 2. We constructed the following automatic protocol:

- First, we separate the two parts, in preparation for comparing them individually to the parts in the score.
- Starting from the first MIDI event, we attempt to identify the corresponding score event by searching for a match within a beat of the next expected score event. This means that if, for instance, at any moment the performer consecutively played wrong notes, or skipped all the notes, during a period of up to one beat, the next correct note would still be properly identified.
- Once the MIDI event is matched to the score event, the time and velocity of the former are written into the `interpretation` array, to produce an entry in the form (index, type, note number, score position, time, velocity) as required.
- If no match in the score is found, then the time and velocity are set to 0.

Each trial of our experiment thus yielded an input interpretation and an output interpretation. We rejected the trials with too many wrong notes, or where the above protocol failed for whatever reason to produce a correct alignment of score and interpretation. We proceeded to an analysis of these data, separately and jointly, in the form of discriminants designed to differentiate between A and H trials.

Piece	Number of trials	μ_Δ	σ_Δ
1	14	0.0656	0.0533
2	14	0.0420	0.0344
3	11	0.0254	0.0219

Table 1. Statistics of desynchronization in the A trials

Piece	Number of trials	μ_Δ	σ_Δ
1	13	0.0872	0.0385
2	13	0.0506	0.0921
3	11	0.0478	0.0684

Table 2. Statistics of desynchronization in the H trials

5.1 Desynchronization

The first question we investigate is whether between the machine and the human, one follows "better", i.e. more tightly. As stated in Sec. 1, our goal in making this model is not to create "perfect accompaniment", but rather "human-like accompaniment". Therefore this question, in the ideal case, should be answered in the negative.

To measure how well-synchronized the two parts are, we examine those notes which are meant to sound together according to the score. That is, we extract subsequences $\{\tilde{s}_k\}, \{\tilde{U}_k\}$ from $\{s_n\}, \{U_m\}$, respectively,including only the elements with indices n, m such that $p_n = P_m$. We define the desynchronization of a given trial as

$$\Delta = \frac{1}{l}\sum_{k=1}^{l} |\tilde{s}_k - \tilde{U}_k|, \tag{2}$$

where l is the number of the synchronization points according to the score (i.e. the number of elements in the sequences $\{\tilde{s}_k\}, \{\tilde{U}_k\}$). In practice, by force of extracting this information from the interpretation arrays, we only took into account synchronization points that were correctly played.

In the A trials, the statistics of Δ are shown in Table 1. The mean μ_Δ and standard deviation σ_Δ are calculated following the removal of trials where the computer suffered from unusually high lag (see Section 7). Similarly for the H trials, Table 2 shows the statistics following the removal of outliers containing too many wrong notes for identifying synchronization points to be practical.

As expected, the average desynchronization is highest for piece 1, as the score contains the least amount of information that a musician can make use of for synchronization. On the other hand, piece 3, with its running 16th notes in the accompaniment, allows the player of the melody to accurately predict the arrival time of the next beat in order to play their corresponding melody note with it.

It is interesting to note that the model, notwithstanding the fact that it was trained on human behavior, was in all cases slightly less desynchronized on average than the human.

Piece	Trials	μ_J	σ_J
1 A	19	0.719×10^3	1.405×10^3
2 A	22	2.817×10^5	2.556×10^5
3 A	24	0.716×10^2	1.355×10^2
1 H	17	3.074×10^2	4.069×10^2
2 H	11	1.205×10^5	0.947×10^5
3 H	17	0.699×10^2	1.546×10^2

Table 3. Statistics of total jerkiness in the A and the H trials

5.2 Jerk

It is sometimes said that machine performances sound jerky when compared to human performances. We tested this hypothesis using a traditional definition of jerk in mechanical movement as the third derivative of position (the second derivative being acceleration).

From the $2N \times 6$ interpretation arrays resulting from human and machine performances, we extracted the columns "time" and "score position" of note onsets, and computed the 3rd-order differences to generate a sequence $j_1, ..., j_{N-3}$ of jerk values. We then defined a measure of total jerkiness:

$$J = \sum_{i=1}^{N-3} j_i^2. \tag{3}$$

Table 3 shows the mean μ_J and the standard deviation σ_J of the total jerkiness in the A trials and the H trials for the three pieces, respectively. We indeed observe that for pieces 1 and 2, the computer accompaniment is jerkier than the human. The fact that this is not the case for piece 3 may be due to the nature of the voice part, which consists mainly of long notes. According to this measure, the model, with its stated goal of accompanying in a human-like manner, performed particularly well in piece 3. In Section 6, we see that A trials of piece 3 were often misidentified as H by participants and listeners alike.

5.3 Velocity Curves

As described at the end of Sec. 3, our model's output follows the velocity of its input according to a running-average-based method. We investigated whether humans, when performing the same task, exhibit a similar behavior. To this end, we supposed the MIDI velocity of each output note of an H trial, that is, each note played by the experimenter, to be a linear combination of the velocities of 11 preceding input notes. We then performed linear regression, with results shown in Figure 4.

We can see that the relationship between the velocities of the two voice parts is much less direct than assumed. For comparison purposes, Figure 4 includes the coefficients for piece 3 in the A setting, which are derived from the fact that since in this part each beat contains 4 notes, our model averages the 4 input velocities received during the previous beat.

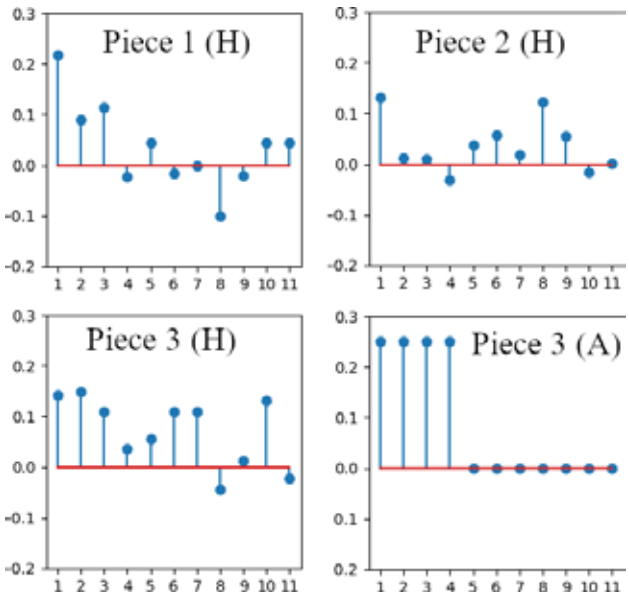


Figure 4. Regression coefficients for velocity prediction

6. SUBJECTIVE EVALUATION

Immediately following each trial, we asked the subject to guess the identity of the partner, and inquired about their reasoning and degree of confidence C (from 0 to 100) in the guess. Separated according to the piece, the results are shown in Table 4.

Piece	Trials	Correct	Success Rate	μ_C
1	28	18	64%	69.5
2	28	22	79%	67.1
3	27	9	33%	59.3

Table 4. Subjects' success rate and their mean confidence in guessing the identity of the partner (A or H)

We asked the same questions to listeners present, with the results shown in Table 5. These tables are tallied by trials and persons jointly; if two listeners were present at a trial, their opinions are counted as if one listener were present at two trials.

Piece	Trials	Correct	Success Rate	μ_C
1	14	12	86%	79.3
2	14	5	39%	75.0
3	10	1	10%	74.0

Table 5. Analogous data to Table 4 for the listeners

In total, 49/83 (59%) of the guesses by the performing participants themselves were correct, vs. 18/38 (47%) by the listeners. However, the listeners had higher average confidence in their guesses than the performing participants.

The fact that both groups' guesses for piece 3 were significantly worse than chance suggests the presence of false expectations. We experienced subjectively our model's performances of the Gounod melody as not sounding "mechanical" in the way that one might be preconditioned to

Piece	μ	max	σ
1	0.0101	0.0620	0.0068
2	0.0105	0.0942	0.0069
3	0.0116	0.2095	0.0088

Table 6. Statistics of the input lag (in seconds)

Piece	μ	max	$\sigma_\triangle$
1	0.0052	0.0508	0.0070
2	0.0053	0.1131	0.0076
3	0.0041	0.0588	0.0061

Table 7. Statistics of the output lag (in seconds)

assume.

Though our focus is on the "musicality" of collaborative performance, other factors inevitably come into play when dealing with responses to such a general question as identifying a partner as human or machine. For instance, the synchronization of key noises to the accompaniment heard, despite our best effort to play along on a disconnected keyboard in A trials, might have served as a clue. The presence or absence of missing or wrong notes in the accompaniment also revealed itself as a telling difference, as errors were unique to the human partners. We tried to minimize this factor by playing our part carefully, and, in the event of being asked by the participant, suggesting that we possibly programmed the model to introduce random errors. Nevertheless, the participants might have drawn upon this rationale all the same to distinguish between our model and the human.

7. DISCUSSION

In the course of setting up our experiment, we determined by trial and error that the highest framerate at which our equipment was able to run the program smoothly was about $f = 100$/sec. That is, the output timing predictions (see Sec. 3) were updated 100 times per second. Note that the input and output timestamps were not quantized, but rather calculated with the maximum precision offered by our MIDI interface.

Nevertheless, even with the program running smoothly, non-negligible sources of lag affected both the input and the output; the program received the input events later than they occurred, and the output sent to the speakers was delayed with respect to the actual calculated output of the program. In the notation of Sec. 4, $t_n > s_n$ and $U_m > T^*_m$.

We measured the input lag $(t_n - s_n)$ and its mean, maximum value, and standard deviation across all trials of a given piece are shown in Table 6. Similarly, the statistics for the output lag $(U_m - T^*_m)$ are shown in Table 7.

As we see, the severity of the lag depended on the rates of input and output notes, the second piece being heavy in output and the third being heavy in input. Due to the construction of the model, input lag mainly affects the refreshing of the predictions (as prompted by the protocol described in Section 3), which while undesirable, is often unnoticeable since with a reasonably consistent tempo the new predictions are usually similar to the old ones. However, output lag has a more noticeable effect, as it directly delays what the listener hears. In the second piece, which requires up to approximately 10 output notes per second, notes were on occasion sounded up to 0.1131 seconds later than intended.

Our experiment subjects frequently commented on the feeling of being slowed down by their partner. It is reasonable to surmise that the lag described here is a significant contributor to this phenomenon.

A priori expectations regarding automatic accompanists play a significant role in experiments such as ours. Not only do they impact subjective evaluations after the experiment, they might also affect how the subjects play during the experiment. We have attempted to separate such effects through our "blind trial" design, in which the partner was chosen at random between a human and a machine.

We have chosen the repertoire pieces primarily based on their prevalence. That is, most people who have learnt the piano are familiar with the works in question, and potential problems of being unable to read or play them fluently were avoided. There exist surely classical works which, in a comparable test situation, would make the differences between human and machine performance more or less apparent. At one end of the spectrum, a piece where the part includes extended and important solos would lead to the performer being recognizable, and in particular, a machine being identified as such. On the other hand, a recitative-like accompaniment might be relatively easy to imitate. Indeed, this idea is upheld by the fact that our participants correctly identified the partner in the piece with the busy accompaniment (Mozart, Variation 2) much more often than in the piece where the roles are reversed (Bach/Gounod).

It is somewhat surprising that although the velocity-based discriminant presented in Section 5.3 identified an obvious difference between human and machine performances, it apparently did not have a bearing on the subjective evaluation by our participants. Further experiments may elucidate whether this aspect might be considered unimportant, or the present results were confused by unclear or diverse expectations. For example, if half of the participants consider that following their dynamic curve is a characteristic machine trait while the other half considers it to be characteristically human, analyzing the data as we have done would yield no result.

Compared with a real musician, our model only makes use of a small portion of the information theoretically available during a collaborative performance. Beyond the knowledge of the notes of a score, it has no intrinsic understanding of its musical content, simply continuing at a constant tempo and dynamic in the absence of continued external input. In fact, it only listens to the timing of its partner and, somewhat crudely, to their dynamics. Other aspects, such as the length of notes (often referred to as "articulation") as well as more complex paradigms of phrasing would appear intuitively desirable for natural-seeming musical collaboration.

8. CONCLUSION

In the context of creating an AI performer in a classical music environment, our model represents an attempt to capture certain aspects of how human musicians interact with each other during an ensemble performance. To this end, we have designed it starting from concepts reflecting our experience as musicians, and adjusted it using data gleaned from accompaniment and collaboration trials. When we set the trained model to play together with external experiment participants, we were able to collect information that allows us to analyze its performance, compared with that of humans recorded under nearly identical circumstances. As part of the experiment, participants and listeners were asked to distinguish, in a blind-trial-like setting, between human and machine. In this paper we presented some measures that quantify salient aspects of the quality of collaboration. A certain number of them, when applied to our data to evaluate the extent to which our model resembles human musicians, produce results that appear to be in accordance with, and perhaps explicative of the participants' responses.

Acknowledgments

This research was supported by the National Science and Technology Council of Taiwan under grant No. 111-2634-F-002-023.

9. REFERENCES

[1] C. Raphael, "A probabilistic expert system for automatic musical accompaniment," *J. Computational and Graphical Statistics*, vol. 10, no. 3, pp. 487–512, 2001.

[2] M. Leman, "Co-regulated timing in music ensembles: A bayesian listener perspective," *J. New Music Res.*, vol. 50, no. 2, pp. 121–132, 2021.

[3] S. Sako, R. Yamamoto, and T. Kitamura, "Ryry: A real-time score-following automatic accompaniment playback system capable of real performances with errors, repeats and jumps," in *Proc. 10th Int. Conf. Active Media Technology*, Warsaw, Poland, 2014.

[4] C. Raphael, "Music plus one and machine learning," in *Proc. 27th Int. Conf. Machine Learning*, 2010, pp. 21–28.

[5] P. Toiviainen, "An interactive MIDI accompanist," *Computer Music J.*, vol. 22, no. 4, pp. 63–75, 1998.

[6] C. Cancino-Chacón, S. Peter, P. Hu, E. Karystinaios, F. Henkel, F. Foscarin, N. Varga, and G. Widmer, "The ACCompanion: Combining reactivity, robustness, and musical expressivity in an automatic piano accompanist," https://arxiv.org/abs/2304.12939, 2023.

[7] E. W. Large, "Resonating to musical rhythm: theory and experiment," in *The Psychology of Time*. Emerald Group Publishing, 2008, pp. 189–231.

[8] Y. Kuramoto, *Chemical Oscillations, Waves, and Turbulence*. New York: Springer-Verlag, 1984.

[9] S. Shahal, A. Wurzberg, I. Sibony, H. Duadi, E. Shniderman, D. Weymouth, N. Davidson, and M. Fridman, "Synchronization of complex human networks," *Nature Comm.*, vol. 11, no. 1, p. 3854, 2020.

[10] K. Armstrong, J.-X. Huang, T.-C. Hung, J.-H. Huang, and Y.-W. Liu, "Real-time piano accompaniment using Kuramoto model for human-like synchronization," in *Proc. 16th Int. Symp. Computer Music Multidisciplinary Research*, Tokyo, Japan, 2023, pp. 744–747.

[11] J. A. Acebron, L. L. Bonilla, C. J. Vicente, F. Ritort, and R. Spigler, "The Kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of Modern Physics*, vol. 77, no. 1, pp. 137–185, 2005.

[12] O. Heggli, J. Cabral, I. Konvalinka, P. Vuust, and M. Kringelbach, "A Kuramoto model of self-other integration across interpersonal synchronization strategies," *PLoS Comput. Biol.*, vol. 15, no. 10, p. e1007422, 2019.

REAL-TIME FUTURE-RHYTHM VISUALIZER FOR DJ PERFORMANCE

Masatoshi HAMANAKA (masatoshi.hamanaka@riken.jp) (0000-0002-0249-4900)[1]

[1]**RIKEN**, Tokyo, Japan

ABSTRACT

In breaking, which features two dancers breakdancing against each other to the rhythm of music remixed by a disc jockey (DJ), it is important to match the technique to the timing of the music beat generated by a DJ. However, it is difficult for many in an audience to judge whether the timing is correct. We developed a real-time future-rhythm visualizer that superimposes in real time visualized video of beats from the past to the future of the music remixed by a DJ on the live broadcast video of breaking. However, the timing of the two does not match due to transmission and equipment delays when the video camera is capturing video and the computer is playing visualization video. Therefore, we made it possible to adjust the timing by changing the position of the object representing the present time. The visualizer was used at the All-Japan Breaking National Championships recorded by NHK, a national TV station in Japan.

1. INTRODUCTION

Breaking, an official competition of the Paris Olympics, features two dancers breakdancing against each other to the rhythm of music remixed by a disc jockey (DJ). The DJ uses two turntables to play multiple musical pieces while switching between them. The DJ then adds accents to the music by scratching the turntable.

Freezing while performing tricks, such as inverted stands, in perfect time with the rhythm generated by the DJ is called "kill the beat" and is the highlight of the competition, as it requires a high level of skill. Most of the audience at a breaking venue have experienced breaking, and they become very excited when they see a successful kill the beat.

However, viewers who are watching breaking for the first time have trouble judging the success of a kill the beat on TV or video. Therefore, we developed a real-time future-rhythm visualizer that displays the future rhythm of the music being remixed by the DJ.

Several methods were proposed to automatically generate a setlist for DJs to prepare before remixing [1–3]. An automated DJ-mixing system was also proposed [4] as well as a method for smoothly crossfading from one piece to another [5].

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

While these studies [1–5] intended to automate the DJ's work or assist the DJ, our study is intended to assist the breaking audience through visualization. Rhythm visualization helps the audience understand if the rhythm matches the timing of the moves. Visualization will also enable the hearing impaired to understand the timing of the rhythm and enjoy breaking.

Our visualizer was used on February 2024 at the All Japan Breaking Championships recorded by the Japan Broadcasting Corporation (NHK), a national TV station in Japan (Fig. 1). The recorded video can be viewed on NHK's website.

Figure 1. Live recording by NHK of our real-time future-rhythm visualizer

2. VISUALIZATION OF FUTURE RHYTHMS

In breaking, dancers improvise to the pieces remixed by the DJ, so it is not disclosed in advance which pieces will be used. Many pieces used by DJs include local tempo change because they make new pieces remixed from old pieces. The tempo of a piece can be changed by operating the DJ mixer. The DJ can also manipulate the disc to temporarily stop a piece or stretch or shorten a sound. Therefore, it is difficult to visualize future rhythms in music remixed by DJs.

2.1 Future Rhythms are Unpredictable

It is difficult to predict future rhythms of music remixed by DJs because changes are added to the music as the disc is manipulated. Therefore, we receive all pieces used by DJs, analyze them, and generate rhythm-visualization source videos. The sound and video are played back in sync by creating a link between the pieces and source video on the DJ software (Fig. 2) [6]. The rhythm-visualization source video consists of long bars representing down beats

and short bars representing up beats, which scroll vertically depending on time. The interval between the bars is constant, and the scrolling speed varies depending on the tempo. When the DJ increases the playback tempo of the music, the playback speed of the video also increases; thus, the speed at which the bars scroll vertically increases.

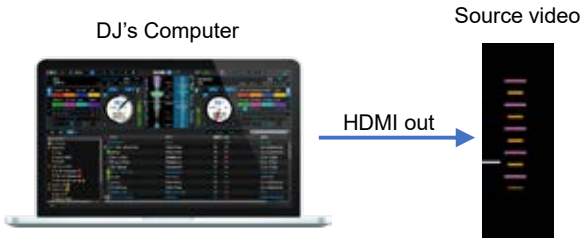


Figure 2. Sound and video are played back in sync

2.2 Appropriate Musical Analysis

The up and down beats can be obtained through artificial intelligence (AI) analysis. However, simple analysis by the AI sometimes results in the up and down beats being reversed, depending on the timbre of the instrument or other factors. Therefore, we added a musical analysis to our visualizer to solve the problem of up and down beats being reversed [7–10].

3. REAL-TIME AND NON-REAL-TIME RHYTHM VISUALIZATION

The broadcast video produced by NHK includes delays as it passes through selectors that integrate multiple cameras and switchers that superimpose information, etc. The distance between the cameras and broadcast room at NHK Hall, the venue of the Championships, was more than 100 meters, which causes transmission delays. Our visualizer also needs to send information from the DJ booth to the broadcast room, which causes delays.

3.1 Adjust Current Timing Indicator

Due to these delays, the timing between the dance video by NHK and the rhythm-visualization video from the DJ's computer is shifted. Therefore, we can adjust the position of the rhythm-visualization output video by adding a line indicating the current time on the source video later in the program so that the timing of the dance video and the rhythm-visualization output video are aligned (Fig. 3).

3.2 Animation for Current Timing

Adding an effect, such as a color change or animation, when the up or down beat bars pass the current timing makes the timing easier for the viewer to understand. However, a video with effects added in advance cannot be created because the timing and position change depending on the delay in the previous item. Therefore, we developed a program for our visualizer that analyzes the rhythm-visualization video being played, reads the bar positions, and adds effects at the appropriate time.

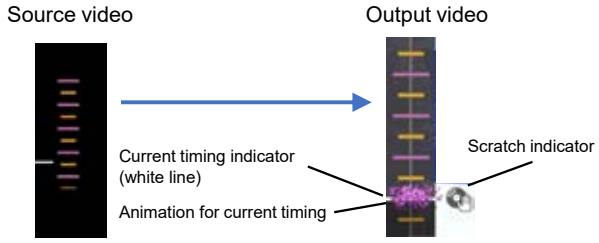


Figure 3. Sound and output video

4. SCRATCHING VISUALIZATION

Two turntables are connected to the DJ mixer, and the DJ software can be used to set the pieces assigned to each turntable. When using the DJ software, a vinyl record, in which sine waves with different phases are recorded on the left and right sides of the vinyl record, is placed on the turntable. The DJ software receives the rotation information of the turntable from the digital vinyl system (DVS) signal generated by the vinyl record. The DJ software then receives the turntable rotation information from the generated DVS signal.

4.1 Detection of Turntable Rotation

Because DVS signals are analog signals, we branch them out, receive them at an audio interface, and analyze them to determine the rotation per minute of both the left and right turntables. Specifically, accelerating the turntable increases the frequency, and decelerating the turntable decreases the frequency. Because the DVS signal is a stereo signal and is out of phase between left and right, it can also detect whether the rotation is forward or reverse depending on which waveform arrives first.

4.2 Detection of Crossfader

DJ mixers have a crossfader, which is used for a swap between two pieces or for adding scratching sounds. Scratching is a technique in which a turntable is operated to rub the other turntable during the playback of a certain piece to emphasize the rhythm with the sound generated. The crossfader can then be moved slightly to the turntable on the side that is being scratched to mix both the playback and scratch sounds.

Many DJs are constantly performing scratching operations on turntables with the crossfader closed and no sound coming out, and they open the crossfader to output the scratching sound when necessary timing. Therefore, to detect scratches, we need to detect not only the rotation of the turntable but also the position of the crossfader.

Many models of DJ mixers that can be connected to a computer via a universal serial bus (USB) cable use Musical Instrument Digital Interface (MIDI) signals for communication. Some DJ mixer models that are equipped with two USB ports for DJ alternation can receive the USB signal flowing through the port in use on the other USB port. Therefore, the MIDI signal is analyzed, and the crossfader position is extracted.

When a scratch is detected, a scratch indicator is displayed. Since the fader moving distance and the rotation speed at the time of a scratch vary greatly from DJ to DJ, the threshold for judging a scratch is manually changed for each DJ.

5. VISUALIZATION LAYOUT AND COLORS

The visualization layout was determined with reference to the NHK broadcast video from the previous year's competition. The entire screen is 1080 pixels in height and 2048 pixels in width, as 2024's broadcast was planned in 2K resolution. The dancers were displayed in the center of the screen, and various displays, such as program names, athlete names, and judgment results, were superimposed on the top and bottom of the screen. Therefore, it was decided to use a vertical display on the left edge so as not to interfere with the displayed dancers.

Of the 550 vertical pixels, approximately 440 are used to display rhythms from the future to the present, and the remainder are used to display past rhythms. The current timing indicator is displayed at the current position but can be moved up 20 pixels and down 20 pixels to absorb the delay. The current portion from the future is brightened and the past portion is darkened to draw attention to the future rhythm. The bar spacing was designed so that four down beats and four up beats are displayed (Fig. 4).

Figure 4. Layout of visualization area

The colors of the bars were selected with reference to the Tokyo Metropolitan Government's universal design guidelines [11], so that even people with diverse color vision can easily distinguish between down beat and up beat by color.

6. GENERATION OF VISUALIZATION SOURCE VIDEO

We received 300 copyright-free pieces from DJs performing at the competition, and generated visualization source videos for all of them.

6.1 Beat-timing Data

Music acoustic signals were analyzed to generate beat-timing data. Since many of the pieces had fluctuating tempos, tempo detection was executed from the audio data using the music-production software Logic Pro's Smart Tempo. For pieces with a beat per minute (BPM) higher than 130, the BPM was multiplied by 0.5. For pieces with a BPM of 64 or less, the BPM was set to 2 times. This is to prevent large changes, such as doubling or halving the speed of the bars in the visualization results, when the DJ smoothly connects from one piece to another.

MIDI messages representing the down-beat and upbeat were then written to each tempo-adjusted measure. When Logic Pro's Smart Tempo analysis did not produce the expected results, manual tempo adjustments were made.

6.2 Visualization Source Video Recorder

Beat-timing data are played back in Logic Pro to output MIDI messages representing the up and down beats and MIDI timing codes. We built a system to read those signals and record visualization source videos.

Figure 1 shows the system configuration. The hardware consists of a combination of non-special equipment. The visualization source video generated by computer (Mac Studio) is output via High-Definition Multimedia Interface (HDMI), which is then recorded by an HDMI capture device connected to the computer via USB-C.

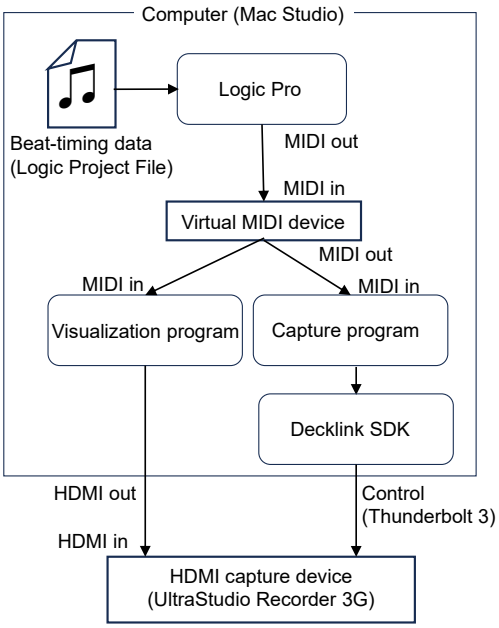


Figure 5. Visualization-source-video recorder

MIDI messages representing the up and down beats played by Logic Pro and MIDI timing codes are sent to the capture and visualization programs we built via a virtual MIDI device, a standard function of Mac OS.

The visualization program displays a bar on the screen when it receives a MIDI message indicating the up or down beat and changes the scrolling speed on the basis of the BPM information received by the MIDI timing code. In other words, the interval between bars is constant and the scrolling speed changes with the BPM.

The capture program starts recording when it receives a MIDI message and stops recording when no more MIDI messages are received.

We tried several HDMI capture devices, but those that execute hardware encoding when capturing video, such

as Blackmagic Design's UltraStudio 4K Extream, do not have a constant delay between receiving the start of recording signal and actual start of recording, making it difficult to synchronize the start timing of the piece and video. However, Blackmagic Design's UltraStudio Recorder 3G, which encodes in software, had almost constant latency, so we decided to use it. The capture devices were controlled using the Decklink software development kit (SDK) from Blackmagic Design [12].

All the visualization source videos were generated in batch processing using Automatar, a standard function of Mac OS.

7. IMPLEMENTATION

Figure 6 shows an overview of our visualizer. We aimed to achieve the visualizer by combining general-purpose equipment without using special equipment. Turntables rotation information is obtained by branching the audio cable that connects the turntable to the mixer and by inputting it to the audio interface connected by a USB cable to a computer (Mac mini). The USB cable is connected between the mixer and Mac mini to acquire the crossfader position. The Mac mini then detects the scratch from the rotation and crossfader information, which is then transmitted to the computer (Mac Studio) via MIDI signals.

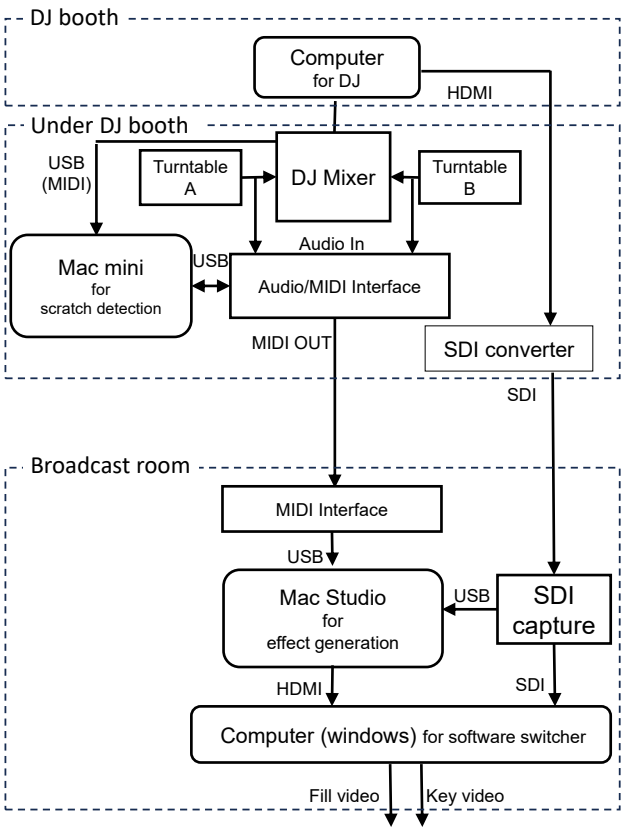


Figure 6. Overview of our real-time future rhythm visualizer

The video of the rhythm-visualization output from the computer used by a DJ is captured with Mac Studio, and image analysis is executed to detect when the bar passes the pixel set as current. Mac Studio then draws the scratch and animation of the bar as it passes.

Finally, the video drawn with Mac Studio and the rhythm-visualization video are captured with a Windows machine to generate the fill video and key video for the video composition.

The distance from the DJ booth to the broadcast room in NHK Hall (Fig. 7) was more than 100 meters, the details of the line between the booth and broadcast room are omitted from Fig. 6. For serial digital interface (SDI) transmission, we used the optical transmission equipment provided by NHK Hall, but it was SDI (3G-SDI) up to HD resolution. Since the output of the SDI converter was 4K-SDI (12G SDI), it was converted from 12G to 3G using an SDI down converter before transmission. MIDI transmission was executed by connecting an small form-factor pluggable (SFP) transceiver to a standard LEMO [13] fiber optic cable for the TV camera to construct a 10G-base network and using Network MIDI, a standard Mac function.

Figure 7. Our equipment in broadcast room in NHK Hall

8. EXPERIMENTAL RESULTS

Our visualizer was used at the Japan Breaking Championships in February 2024. Since the superimposed visualization video will be recorded live, we carefully checked the visualization source video in advance. Absorption of transmission delays was adjusted on site as it depended on the hall's equipment.

8.1 Video Screening

Using the system described in a previous study 6.2, 300 visualization source videos were exported in batch processing. The exported videos were manually screened, and half had problems. There were two causes of these problems: a timing error at the start of recording and image distortion. For the timing error, there was a blank space at the beginning of the video or the video started in the middle. For image distortion, the color of the bars was not uniform and there were unnecessary lines. After repeated exporting of the images with abnormalities and manual screening, all images with no abnormalities were generated after the eighth export.

8.2 Timing Screening

Although the timing of the exported visualization source video should be correct in principle, it was manually verified: 10 arbitrary pieces were selected out of the 300 pieces, and the visualization source video of the piece was tacked onto Logic Project and visually verified by a music and video expert. We confirmed that the timing was correct for all ten pieces, even after slow playback. For ten different pieces, the DJ software Serato [6] was used to create a link between the song and video, and the timing was visually checked while changing the BPM.

8.3 Absorption of Transmission and Equipment Delays

The video received from NHK was switched from multiple camera synchronized by time code. However, the sound on stage, the emcee's voice on stage, and the sounds remixed by the DJ were recorded by microphones. Synchronization of the video and sound was lip-synced by NHK using the voice and video of the emcee during rehearsals. The video was recorded after the rhythm-visualization video was superimposed on that video. When the recorded video was checked during slow playback, it was found that the visualization video was delayed by two frames at 59.94 frames per second. We then moved the current timing indicator down 20 pixels and re-recorded it; therefore, the visualized video and sound from NHK were synchronized.

9. CONCLUSIONS

We developed a real-time future-rhythm visualizer to visualize the music remixed by DJs in real time. The visualizer was used at the All-Japan Breaking National Championships and recorded by NHK, a national TV station in Japan. The recorded video can be viewed on NHK's website.

`https://www.nhk.jp/p/ts/9WXX8RKJVY/movie/` (access from Japan only).

When 20 people who viewed the recorded video were asked for their opinions, 18 said they enjoyed watching it, and 14 said "Kill the beat" was easy to understand.

The data generation and screening process involved a large amount of manual work, which we plan to pipeline and automate. In visualization-source-video export, there is currently a trade-off between timing error and image distortion. HDMI capture with software encoding has fewer timing errors but more image distortion. However, HDMI capture with hardware encoding has more timing errors but less image distortion. We plan to investigate ways to reduce both types of errors by using multiple machines and conduct more detailed experiments.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 24H00748 and 21H03572.

10. REFERENCES

[1] R. M. Bittner, M. Gu, G. Hernandez, E. J. Humphrey, T. Jehan, H. McCurry, and N. Montecchio, "Automatic playlist sequencing and transitions." in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR2017)*, 2017, pp. 442–448.

[2] T. Hirai, H. Doi, and S. Morishima, "Musicmixer: Automatic dj system considering beat and latent topic similarity," in *Proceedings of 22nd International Conference of MultiMedia Modeling (MMM2016)*. Cham: Springer International Publishing, 2016, pp. 698–709.

[3] J. Parera, "Dj codo nudo: a novel method for seamless transition between songs for electronic music," *Master's thesis, Universitat PompeuFabra, Barcelona*, 2016.

[4] Y. T. Hiromi Ishizaki, Keiichiro Hoashi, "Full-automatic dj mixing system with optimal tempo adjustment based on measurement function of user discomfort," in *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR2009)*, 2009, pp. 135–140.

[5] K. Robinson and D. Brown, "Automated time-frequency domain audio crossfades using graph cuts," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR2019) Late-Breaking/Demo*, 2019.

[6] Serato, "Serato DJ Pro - Professional DJ Software - Download — serato.com," https://serato.com/dj/pro, [Accessed 01-03-2024].

[7] M. Hamanaka, K. Hirata, and S. Tojo, "Implementing a generative theory of tonal music," *Journal of New Music Research*, vol. 35, no. 4, pp. 249–277, 2006. [Online]. Available: https://doi.org/10.1080/09298210701563238

[8] M. Hamanaka, K. Hirata, and S. Tojo, "Automatic generation of metrical structure based on gttm," in *Proceedings of the 2005 International Computer Music Conference (ICMC2005)*, 2005, pp. 53–56.

[9] M. Hamanaka, K. Hirata, and S. Tojo, "deepGTTM-I&II: Local Boundary and Metrical Structure Analyzer Based on Deep Learning Technique," in *Computer Music Modeling and Retrieval*, 2016. [Online]. Available: https://api.semanticscholar.org/CorpusID:42476381

[10] M. Hamanaka, K. Hirata, and S. Tojo, "DeepGTTM-II: Automatic Generation of Metrical Structure Based on Deep Learning Technique." Proceedings of 13th Sound and Music Computing Conference (SMC2016), Aug. 2017. [Online]. Available: https://doi.org/10.5281/zenodo.851229

[11] Tokyo Metropolitan Government Bureau of Social Bureau of Social Welfare, “Tokyo metropolitan government color universal design guidelines,” https://www.fukushi.metro.tokyo.lg.jp/kiban/machizukuri/kanren/, [Accessed 01-03-2024](in Japanese).

[12] Blackmagic Design, “DeckLink ; Software | Blackmagic Design — blackmagicdesign.com,” https://www.blackmagicdesign.com/products/decklink/software, 2013, [Accessed 02-03-2024].

[13] LEMO, “About us | LEMO Connectors and cables — lemo.com,” https://www.lemo.com/our-world/company/about-us, [Accessed 07-03-2024].

SMUG-EXPLAIN: A FRAMEWORK FOR SYMBOLIC MUSIC GRAPH EXPLANATIONS

Emmanouil KARYSTINAIOS (emmanouil.karystinaios@jku.at) [1], **Francesco FOSCARIN**[1,2], and **Gerhard WIDMER**[1,2]

[1] *Institute of Computational Perception*, **Johannes Kepler University Linz**, Austria
[2] *LIT AI Lab*, **Linz Institute of Technology**, Austria

ABSTRACT

In this work, we present Score MUsic Graph (SMUG)-Explain, a framework for generating and visualizing explanations of graph neural networks applied to arbitrary prediction tasks on musical scores. Our system allows the user to visualize the contribution of input notes (and note features) to the network output, directly in the context of the musical score. We provide an interactive interface based on the music notation engraving library Verovio. We showcase the usage of SMUG-Explain on the task of cadence detection in classical music. All code is available on `https://github.com/manoskary/SMUG-Explain`.

1. INTRODUCTION

In recent years, Graph Neural Networks (GNNs) have emerged as a method for processing musical scores in Music Information Research (MIR) applications, such as cadence detection [1], expressive performance rendering [2], optical music recognition [3], music generation [4], voice separation [5], and Roman numeral analysis [6]. Like the majority of deep learning-based approaches, GNNs are not intrinsically interpretable, thus making it impossible to inspect the system to reveal potential issues of the system itself and the data it uses or to gain knowledge about the specific task [7]. One could modify the system to make it more interpretable, but this often leads to a reduction in performance. A popular alternative is the so-called *post-hoc* method which aims to explain already trained deep models.

In the field of MIR, multiple explanation techniques have been proposed in recent years [8–12], but they all target systems which use matrix-like inputs, such as spectrograms or pianorolls. In this paper, instead, we focus on the explainability of MIR systems that use graphs as input and process musical scores. We argue that graph explanations for scores are more musically interpretable since they point to individual note elements and their neighborhood in the score. We experiment with various post-hoc gradient-based explanation methods for GNNs from the literature [13–19]. The generated explanations are quantitatively evaluated by verifying that our explanations satisfy the sufficiency and necessity conditions from a GNN point of view [18]. We name our framework Score MUsic Graph (SMUG)-Explain.

Once the explanations are produced, there remains the question of how to present them to the user in an effective way. We display them directly on the musical score, to promote a clear and intuitive relation between the system input, the output and the explanations. Moreover, our use case is more complex than the (more common) explanation of global classifiers (i.e., systems that predict one label for each input excerpt). We want to be able to target systems that predict labels for multiple elements in the input, e.g., for every note or for every time step. This calls for an interactive interface that enables users to focus on specific subsets of predictions. We start with a web-based graphical rendition of a digitized musical score, based on the engraving library Verovio [20]. The user can interact with individual notes to trigger the visualization of explanations of the target GNN model. Specifically, each note is associated with the subgraph that most contributes to the underlying model's prediction. Furthermore, for each selected note, the user can visualize its feature importance, i.e. the relative importance of input features of the notes involved in the explanation.

We showcase SMUG-Explain on the cadence detection model of Karystinaios et al. [1], by comparing different explanation techniques, and by a qualitative analysis of three music excerpts. GNNs excel at capturing intricate relationships and patterns within graph-structured data. In music, that could be related to voice-leading, harmonic relations, melodic patterns, and other elements that are implicitly modeled by the score graph. From a musicological point of view, some parallelism can be found between our explainer and an analyst who highlights the voice leading and most important notes that contribute to the analysis assessment. In particular, some analysis methodologies, such as Schenkerian analysis or GTTM [21], also create graph structures between notes. In the second part of the paper, we attempt to shed some insight into more musical interpretations of graph explanations. To achieve this, we apply our framework to several music excerpts and comment on the generated explanations. We believe that explanations of deep analytical models might contain some musical pointers that can correlate with expert musical analyses.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. PRELIMINARY CONCEPTS

In this section, we present a common task-independent approach (as emerging from recent papers on the topic) to modelling musical scores with GNNs, then we give some information on explainability techniques for GNNs and their evaluation.

2.1 GNN-based Approaches on Musical Scores

The fundamental idea of GNN-based approaches to musical scores is to model a musical score as a graph where notes are the vertices and edges model the temporal relation between the notes. [1] The most common approach [1,2,5,6] to create a graph from a musical score considers four types of edges (see Figure 1 for visualization on the score):

- *onset edges*: connect notes that share the same onset;
- *consecutive edges* (or *next edges*): connect a note x to a note y if the offset of x corresponds to the onset of y;
- *during edges*: connect a note x to a note y if the onset of y falls within the onset and offset of x;
- *rest edges* (or *silence edges*): connect the last notes before a rest to the first ones after it.

The GNN can treat these four edge types with a single representation [1], thus considering a *homogeneous* input graph, or can treat them as different representations per edge type, i.e. as a *heterogeneous* graph.

Adopting the latter approach, a score graph is represented as an attributed heterogeneous graph $G = (V, E, \mathcal{R}, X)$, where V is the set of nodes representing the notes in a score. E is the set of typed edges with elements of the form (v, τ, u) where, $v, u \in V$ and $r \in \mathcal{R}$ is a relation type. Finally, $X \in \mathcal{R}^{V \times k}$ is a feature matrix such that every node u has its corresponding feature vector $x_u \in X$. Furthermore, we additionally can construct an adjacency matrix $A \in V \times V$.

In our work, we apply the GraphSAGE convolutional block [22]. For a node v the features message passing process per layer l is described as follows:

$$\begin{aligned} h_{\mathcal{N}(v)}^{(l+1)} &= \text{aggregate}\left(\{h_j^l, \forall j \in \mathcal{N}(v)\}\right) \\ h_v^{(l+1)} &= \sigma\left(W \cdot \text{concat}(h_v^l, h_{\mathcal{N}(v)}^{l+1})\right) \\ h_v^{(l+1)} &= \text{norm}(h_v^{(l+1)}) \end{aligned} \tag{1}$$

Where W is a learnable weight, $\mathcal{N}(v)$ is the set of neighbors of v, aggregate is a permutation invariant aggregation function such as mean, sum, or max, and σ is an activation function such as ReLU.

[1] More elements could be used as vertices, such as rests, bar lines, and dynamics symbols, but these are not commonly used and we do not consider them in this paper.

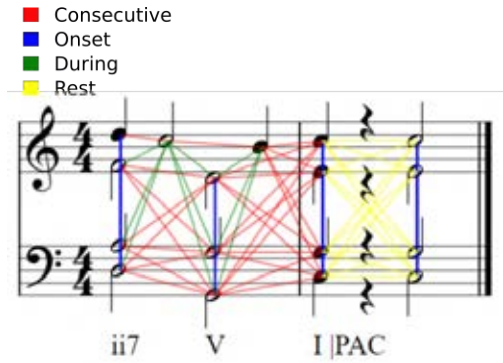


Figure 1. An example of a score graph depicting the different graph edge types in different colours.

2.2 Explainability and Graphs

As explained in the introduction, we are interested in *post-hoc* methods, i.e., explainability techniques that work on pre-trained models. Within the post-hoc realm, the dichotomy of model-aware and model-agnostic explanations emerges. Model-aware methods dissect model parameters for insights, while model-agnostic approaches treat the model as a black box, perturbing inputs to unveil the significance of elements in the output. In this work, we try both categories but find the model-aware techniques to work better for our case.

In terms of explanation evaluation, post-hoc GNN explanations can be measured using the *fidelity metric* [19]. The fidelity metric measures the impact of the generated explanatory subgraph on the initial prediction, achieved either by exclusively presenting the subgraph to the model (fidelity-) or by excluding it from the entire graph (fidelity+). These fidelity scores capture the ability of an interpretable model to replicate the intrinsic logic of the natural phenomenon or the GNN model.

When it comes to describing post-hoc explanations, we can identify two types of explanations based on their fidelity scores: *necessary* and *sufficient*. A sufficient explanation can be used on its own to reproduce the model's prediction, and it gets a near-zero negative fidelity score. However, a sufficient explanation is not necessarily unique. On the flip side, a necessary explanation is crucial – removing it from the initial graph changes the model's prediction, like a counterfactual explanation. This type of explanation earns a positive fidelity score close to 1. The ideal situation is found when an explanation is both necessary and sufficient. Amara et al. [18] propose to balance the sufficiency and necessity requirements with the *characterization score* which is the weighted harmonic mean of the positive and negative fidelities. Therefore a characterization score close to 1 suggests clear, comprehensive, and informative insights into the model's decisions.

In this work, we use the characterization score as a means to evaluate different explanation techniques and select the most fitting. It has to be noted that the evaluation of explainability techniques is a particularly complex field, and many approaches have been proposed and then put into question by subsequent research [23–25]. Be that as it

may, in this work, it is assumed that the characterization score is a suitable metric to measure the quality of graph explanations.

3. OUR APPROACH

In this section, we detail the cadence detection model that was trained and used to showcase the explanations, then we describe our framework, and finally, we focus on the choice of explanation techniques.

3.1 Cadence Detection Model

To showcase our framework, we chose to use a slightly modified version of the cadence detection model introduced by Karystinaios and Widmer [1]. We extended the model to use as input a heterogeneous score graph as described in Section 2.1. Furthermore, we extended the prediction capabilities of the model from binary (i.e. no-cad or PAC) to multiclass cadence prediction, covering PAC, IAC, and HC. Moreover, we modified the architecture by adding an onset regularization module that sums the latent representations (after the GNN encoder) of all the notes that occur on a distinct onset of the score to every note that shares this onset. In summary, our cadence detection model consists of a graph convolutional encoder, an onset regularization module, an embedded SMOTE layer for training, and a shallow MLP classifier.

During training, the graph input is passed first through the graph encoder. The obtained node embeddings are then grouped by onset based on the score information, and their representations are averaged together. Following this step, embedded SMOTE is applied to balance the number of cadence classes compared to the notes not having cadence labels in the score. However, when doing inference, the latter synthetic oversampling step is skipped. Finally, the oversampled embeddings are given as input to a shallow 2-layer MLP classifier that predicts the cadence type.

We trained our model with a joined corpus of cadence annotations from the DCML corpora [2], the Bach fugues from the well-tempered clavier No. 1 [26], the annotated Mozart string quartets [27], and the annotated Haydn string quartets [28]. Our joined corpus makes for $590,149$ individual notes and $17,188$ cadence annotations. We train our model on 90% of the data and evaluate on 10% using a random split. Our model reaches a mean F-score of 59% on the test set. Note that these results cannot be directly compared with [1], since we use a different (bigger) dataset and perform multiclass prediction.

3.2 The SMUG-Explain Framework

Our framework has two main functionalities: generating explanations and making them visually interpretable.

The first step involves importing musical scores, creating the graph structure, and running the explanation techniques. The score import uses the Python library PARTITURA [29] which supports a variety of score formats, such as MEI, MusicXML, Kern, or (quantized) MIDI. The graph creation is based on previous work [5, 6] and outputs a graph in the widely used PYTORCH GEOMETRIC format [30] to favour reusability and extensibility of our frameworks.

For the explanation part, we follow a standardized GNN model explanation pipeline: the explainer receives a pre-trained GNN model that performs node-level classification and produces an explanation of the cadence label prediction for a specific note, at two levels: (a) by quantifying the relative importance of the various features of the note; and (b) by identifying an explanation subgraph consisting of the notes and their relations that seem most important for the prediction. For producing an explanation subgraph, the explainer computes importance masks over all edges. These importance masks reflect an importance score for each one of the edges in the input graph that is used to filter which (most important) edges belong to the explanation subgraph. The importance masks can be soft (i.e. continuous numbers between 0 and 1) or hard (i.e. binary). Furthermore, a maximum number of edges can be imposed per edge type to limit the explanation to stay within a certain graph size. It has been shown that hard masks tend to increase the necessity and sufficiency of explanations [18]. Therefore, in our application, we use a hard top-k method for each one of our edge types, where k is set to 10. Likewise, for producing the feature importance, the explainer produces masks over all nodes and their features and it keeps the k most important by summing the masks along the feature dimension.

The interface of the SMUG-Explain framework is a web-based interface implemented in HTML and Javascript. The core of the interface is the score engraving library VEROVIO [20] which outputs an SVG representation from a musical score file. We then extend this score representation with the target deep learning model output (i.e., predicted cadences in our case) and with the information produced by the explainers. From the latter, we display in particular the edges between different notes that contribute to the explanation and the feature importance for each note that corresponds to a cadence prediction (see the next section for details). To make this part possible we need a one-to-one mapping between elements in Python and elements in the SVG generated image. Verovio preserves the mapping between note-ids in the musical score file and the SVG image (if they exist), but only the MEI format contains note-ids. Therefore we extended PARTITURA with an MEI export function, and we support Roman numeral analysis and cadence name exports.

The final step for an effective interface consists of making it interactive. The user can click on single notes, and this will trigger the visualization of the corresponding explanation edges and feature importance. Moreover, the user can switch back and forth between the visualization of the input edges (from the graph presented in Section 2.1) and the explanations, and listen to an audio rendition of the musical score. The interface is shown in Figure 2.

3.3 Choice of Explainability Techniques

To investigate and evaluate the explanations produced by our framework we test several explanation algorithms

[2] https://github.com/DCMLab/dcml_corpora

Figure 2. A demonstration of the SMUG-Explain Web interface. In this example, we view the first bars of Mozart's Piano Sonata K280 2nd mvt. It includes a Roman numeral analysis and the cadence label predicted by our model at the top. The purple dashed lines are the produced explanation for the note highlighted in red. Note the vertical connection line in the very first bar, which is also a part of this explanation. At the bottom, we can view the feature importance for the explained note.

such as Saliency, Integrated Gradients, Deconvolution, and Guided Backpropagation. Saliency gauges node and edge importance by weighing each node after calculating the gradient of the output concerning input features [16]. Integrated Gradient tackles the saturation issue of gradient-based methods like Saliency by accumulating gradients along the path from a baseline input (zero-vector) to the current input [31]. Deconvolution computes the gradient of the target output but overrides ReLU function gradients, only propagating non-negative gradients [14]. Guided Backpropagation follows a similar approach, backpropagating only non-negative gradients [13].

Each of those methods was evaluated on model-level explanations, i.e. the explanation algorithm computes its losses with respect to the model output. We use the characterization score as defined in [18] (see Section 2.2). We fix the positive and negative fidelity weights to 0.5 and apply k-top hard masks for edges and nodes. The results are shown in Table 1. Our findings suggest that for this task the Integrated Gradients method better captures explanations, in some cases achieving a perfect characterization score of 1.

We note that most methods tested on our application are gradient-based. Some perturbation-based methods for generation GNN explanations such as the GNNExplainer [32], Occlusion [17], or the GraphMaskExplainer [33], might be suitable for our application, however, they are not yet adapted to work with heterogeneous graphs.

As a general remark, we would like to underline that typically graph-dedicated explainers are more biased toward generating compact explanation subgraphs. However, this bias might be less suitable for music where some analysis methods suggest that important elements of a piece might not be directly interconnected.

Pieces/Model	SAL	GBP	DC	IG
WTC-I Fuga 1	0.0588	0.4706	0.2941	**1.0**
WTC-I Fuga 2	0.0	0.0435	0.0435	**0.9130**
WTC-I Fuga 5	0.0	0.0667	0.0	**0.8667**
Mozart K280-2	0.0	0.3125	0.3438	**0.9375**
Chopin Op.48	0.0	0.1875	0.1250	**0.8125**
Mozart K331	0.0	0.0	0.0625	**1.0**

Table 1. Characterization score for the model explanations of cadences per piece. The four methods are mentioned in Section 3.3. SAL stands for Saliency, GBP for Guided Backpropagation, DC for Deconvolution and IG for Integrated Gradients. Highlighted values indicate the highest (best) explanations in terms of characterization score.

4. QUALITATIVE ANALYSIS

In this section, we perform a qualitative analysis of a diverse set of scores and comment on the graph explanations produced by our system on the task of cadence classification. Apart from the explanation metrics described in Section 3.3 we believe that explanation should be also motivated in musical terms. Therefore, we go into depth about some individual explanations of cadence predictions, targeting both true positive and false positive predictions. [3]

We showcase explanations in four classical music excerpts ranging from the Baroque to the Romantic era. Furthermore, we consult expert analyses in terms of harmony, voice leading, and Schenkerian analysis and attempt to interpret the produced explanations. Naturally, we assert that the explanations discussed should be necessary and sufficient. We remind the reader that the generated explanations do not explain the working mechanics of cadences but rather present insights into the model's decision process. In other words,

[3] One could also focus on no-cadence predictions (and false negatives), but we are not considering these in this section, since the "absence of a specific evidence for a cadence" may result in less musicologically interesting graph patterns.

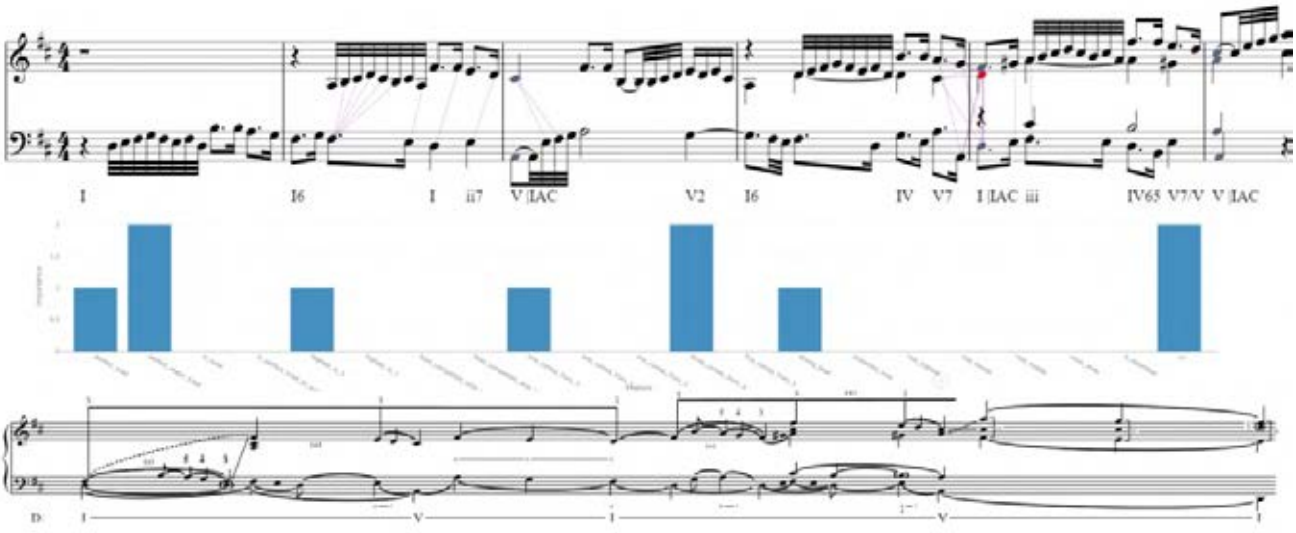


Figure 3. The first bars of the Fuga No. 5 of the Well-Tempered Clavier book No. 1. On the top the score and the explanation of the wrong prediction of the highlighted note in red. In the middle, the feature importance is visualized for the highlighted note. On the bottom, a Schenkerian analysis of the segment by [34]

we are not answering the question "why is there a cadence here?", but "why does the model think there is a cadence here?".

All of the below explanations are included in our publicly available code, including the entire pieces. We invite the reader to explore our interface and test our framework.

4.1 Mozart Piano Sonata K280 Mov. 2

Our first example (see Figure 2) focuses on an excerpt from Mozart's Piano Sonata K280 in F major, from the second movement. The harmonic analysis of this segment was provided by Hentschel et al. [35]. We focus on the perfect authentic cadence arriving on the second beat of measure eight and signaling the end of the first phrase. The harmonic analysis indicates a textbook preparation of the cadence with the German augmented sixth chord on the second beat of measure 6, then the dominant with a tonic in second inversion resolving to a dominant seventh. Finally, the tonic bass arrives on the first beat of measure 8 but with sustained soprano, alto, and tenor voices which themselves resolve on the second beat of the same measure.

We take a closer look at the explanation subgraph of the bass of the tonic (F4) which arrives earlier than the cadential arrival point of the soprano. However, the explanation highlights the descending Urlinie melody containing the $\hat{3}, \hat{2}, \hat{1}$ in the top voice, capturing the first appearance of the $\hat{3}$ on the relative strong beat of the sixth measure (subdominant space), whilst considering the overarching bass arpeggiation of the $i - V - i$. Interestingly, the explanation subgraph also contains the first chord of the piece (see faint vertical line in bar 1), indicating that the model considers some information regarding the key of the piece. It's important to highlight that the cadential ground truth obtained from [35] designates notes with the cadence label at the cadential arrival point of the soprano. However, we argue that including the first appearance of the bass note on the strong beat of the same measure is a crucial addition to the cadence annotation ground truth from a musicological standpoint. This addition is accurately captured by our model. Upon analyzing the predictions, we observe that the model correctly identifies the individual notes involved in the PAC in measure 8, namely the $F4$ on the first beat and the soprano, tenor, and alto resolutions on the second beat (remember that our model predicts cadence labels for individual notes, not score onsets).

Furthermore, we deduce from the explanation subgraph that the next notes after the cadence are also crucial for the model's prediction, as they point towards an ending of a phrase and a new rhythmical and harmonic idea further on.

4.2 Bach WTC Fugue

For this example, we explore the predictions and explanations of cadences in the fifth Fuga in D major from J.S. Bach's Well-tempered Clavier. We believe that the contrapuntal nature of this piece could provide insightful hints about voice leading. Our analysis is complemented by a Schenkerian analysis conducted by Marlowe [34], and we enhance clarity by performing a Roman numeral analysis on the explanatory excerpt (refer to Fig. 3).

Our specific focus centers on the explanation of a false positive IAC (Imperfect Authentic Cadence) prediction on the downbeat of the fifth measure. The ground-truth cadence annotations were provided by [26]. The cadence annotations, following the ground truth, were originally provided by Giraud (2015). IAC labels were annotated on the downbeats of measures 3 and 6, both corresponding to the dominant of the D major (i.e. the tonic). Our model accurately predicts these labels but also anticipates an additional IAC label on the downbeat of measure 5, which is not present in the ground truth annotations. Consequently, we investigate what led to this prediction.

We observe that the explanation subgraph for the high-

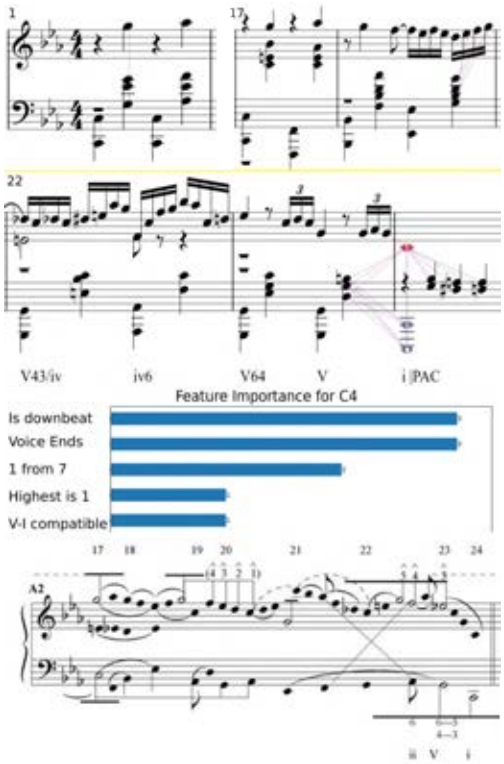


Figure 4. Excerpt of Nocturne Op. 48, no. 1 in C minor by F. Chopin. Top: excerpts of the explanation for Cadence on measure 24. Bar numbers are notated to the top left of each score segment. Middle: Feature importance for the highlighted C4 note in red. Bottom: Middleground voice leading analysis (from [36]).

lighted red note at the top of Figure 3 contains, as expected, the edges between the dominant and the subdominant. However, the interesting part of the subgraph takes place in bars 2-3 of the score. The high voice contains the descending melodic line towards the leading tone which then re-appears right before measure 5, and the highlighted red note subsequently resolves it. The lower voice on measure 2 focuses around the E note which would be a fifth from the dominant A. Therefore, the subgraph outlines a movement ii-V-I for the bass.

The middle-ground Schenkerian analysis displayed in the bottom part of Figure 3 provides some similar observations. We observe an oscillating bass from the tonic to the dominant and over again. This oscillating bass is captured in the cadence annotations by the two cadences on the dominant. It could be argued that the falsely predicted cadence on the tonic strengthens the analysis assumption of the oscillating bass.

In terms of feature importance, we see that characteristics of imperfect cadence are activated such as the existence of a perfect major triad, the highest note being a transposed third interval for the bass, the existence of a leading tone resolving, and the presence of a dominant seventh chord before. Naturally, such characteristics are not exclusively present in the event of cadences but in this case, they seem to influence the model's prediction.

4.3 Chopin Nocturne in C minor op. 48

For our last example, we chose a passage with more ambiguous cadential and harmonic elements: bars 22-24 from Chopin's Nocturne op. 48 no 1, in C minor. For this piece, we consult a Schenkerian analysis by Swinkin [36]. In particular, we look closely at the perfect authentic cadence that arrives in measure 24. Unlike previous examples, this cadence does not have the textbook voice leading and harmonic elements that distinctly define PACs but, nevertheless, carries a cadential character.

Our model correctly identifies the PAC that arrives on the downbeat of measure 24. To provide support for our analysis and commentary, we perform a Roman numeral analysis on the segment depicted in Figure 4. From the model's explanation subgraph, we see that once again the explanation contains the first chord of the piece, further strengthening our assumption that the model is gathering information about the key to inform its predictions. However, the rest of the generated explanation subgraph is mostly compact and local, focusing only on the last notes before the cadence and the ones after it. The chordal content of the segment contains the supported harmony for a perfect authentic cadence (PAC) preparation. But, since the cadence does not follow the typical voice-leading patterns usually involved, it seems that the model does not need to go towards far neighborhoods in the graph to infer the necessary context for its prediction.

Interestingly, the model includes in the explanation subgraph a part of measure 18. The connected notes in measure 18 correspond to a part of the descending Urline that is present in the voice-leading analysis excerpt in the Figure, however, they are not in order. This could be a connection with what one could view as a "sustained" high G in the melody line which slowly descends towards G an octave lower before the cadence. That being said, it is rather ambiguous how the part in measure 18 affects the model's prediction.

5. CONCLUSION AND FUTURE WORK

This paper presented the SMUG-Explain framework for generating and visualizing graph explanations on musical scores. We showcased the framework on a cadence detection model, compared different explanation techniques, and gave some qualitative insights into the explanations.

Future work will focus on developing new explanation techniques dedicated to musical score graph data, and on testing our results with user-based evaluations. The final objective would be to produce a musicologically trustworthy and user-friendly framework that can support expert analysts to produce more effective musical analyses. Furthermore, we aim to invest efforts into making the system more accessible by releasing an online server-based version of our interface, which can be used to produce predictions for some reference models and tasks and explain them, without the need to run any code locally.

6. ACKNOWLEDGEMENTS

This work is supported by the European Research Council (ERC) under the EU's Horizon 2020 research & innovation programme, grant agreement No. 101019375 (*Whither Music?*), and the Federal State of Upper Austria (LIT AI Lab).

7. REFERENCES

[1] E. Karystinaios and G. Widmer, "Cadence detection in symbolic classical music using graph neural networks," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

[2] D. Jeong, T. Kwon, Y. Kim, and J. Nam, "Graph neural network for music score data and modeling expressive piano performance," in *Proceedings of the International Conference on Machine Learning (ICML)*. PMLR, 2019, pp. 3060–3070.

[3] A. Baró, P. Riba, and A. Fornés, "Musigraph: Optical music recognition through object detection and graph neural network," in *International Conference on Frontiers in Handwriting Recognition*. Springer, 2022, pp. 171–184.

[4] E. Cosenza, A. Valenti, and D. Bacciu, "Cosenza, emanuele and valenti, andrea and bacciu, davide," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[5] E. Karystinaios, F. Foscarin, and G. Widmer, "Musical Voice Separation as Link Prediction: Modeling a Musical Perception Task as a Multi-Trajectory Tracking Problem," in *Proceedings of the Joint Conference on Atrificial Intelligence (IJCAI)*, 2023.

[6] E. Karystinaios and G. Widmer, "Roman Numeral Analysis with Graph Neural Networks: Onset-wise Predictions from Note-wise Features," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[7] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022. [Online]. Available: https://christophm.github.io/interpretable-ml-book

[8] S. Mishra, B. L. Sturm, and S. Dixon, "Local Interpretable Model-agnostic Explanations for Music Content Analysis," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 537–543.

[9] S. Mishra, E. Benetos, B. L. Sturm, and S. Dixon, "Reliable Local Explanations for Machine Listening," in *Proceedings of the 2020 International Joint Conference on Neural Networks, IJCNN*. IEEE, 2020, pp. 1–8.

[10] V. Haunschmid, E. Manilow, and G. Widmer, "audioLIME: Listenable Explanations Using Source Separation," in *Proceedings of the International Workshop on Machine Learning and Music, MML*, 2020, pp. 20–24.

[11] A. B. Melchiorre, V. Haunschmid, M. Schedl, and G. Widmer, "LEMONS: Listenable Explanations for Music recOmmeNder Systems," in *Advances in Information Retrieval: Proceedings of the European Conference on IR Research, ECIR*, vol. 12657. Springer, 2021, pp. 531–536.

[12] F. Foscarin, K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Concept-based techniques for "musicologist-friendly" explanations in a deep music classifier," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:251881711

[13] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Workshop at International Conference on Learning Representations (ICLR)*, 2015.

[14] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part I 13*. Springer, 2014, pp. 818–833.

[15] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *The Journal of Machine Learning Research*, vol. 11, pp. 1–18, 2010.

[16] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," in *Workshop at International Conference on Learning Representations (ICLR)*, 2013.

[17] L. Faber, A. K. Moghaddam, and R. Wattenhofer, "When comparing to ground truth is wrong: On evaluating GNN explanation methods," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 332–341.

[18] K. Amara, R. Ying, Z. Zhang, Z. Han, Y. Shan, U. Brandes, S. Schemm, and C. Zhang, "Graphframex: Towards systematic evaluation of explainability methods for graph neural networks," *In Learning on Graphs Conference (Proceedings of Machine Learning Research)*, vol. 198, 2022.

[19] H. Yuan, H. Yu, S. Gui, and S. Ji, "Explainability in graph neural networks: A taxonomic survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 5, pp. 5782–5799, 2022.

[20] L. Pugin, R. Zitellini, and P. Roland, "Verovio: A library for engraving mei music notation into svg," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, 2014.

[21] F. Lerdahl and R. S. Jackendoff, *A Generative Theory of Tonal Music, reissue, with a new preface*. MIT press, 1996.

[22] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[23] K. Hoedt, V. Praher, A. Flexer, and G. Widmer, "Constructing adversarial examples to investigate the plausibility of explanations in deep audio and image classifiers," *Neural Computing and Applications*, vol. 35, no. 14, pp. 10 011–10 029, 2023.

[24] J. Adebayo, J. Gilmer, M. Muelly, I. J. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Neural Information Processing Systems*, 2018.

[25] G. Yona and D. Greenfeld, "Revisiting sanity checks for saliency maps," in *Workshop on eXplainable AI approaches for debugging and diagnosis (XAI4)*, 2021.

[26] M. Giraud, R. Groult, E. Leguy, and F. Levé, "Computational fugue analysis," *Computer Music Journal*, vol. 39, no. 2, pp. 77–96, 2015.

[27] P. Allegraud, L. Bigo, L. Feisthauer, M. Giraud, R. Groult, E. Leguy, and F. Levé, "Learning sonata form structure on mozart's string quartets," *Transactions of the International Society for Music Information Retrieval (TISMIR)*, vol. 2, no. 1, pp. 82–96, 2019.

[28] D. R. Sears, M. T. Pearce, W. E. Caplin, and S. McAdams, "Simulating melodic and harmonic expectations for tonal cadences using probabilistic models," *Journal of New Music Research*, vol. 47, no. 1, pp. 29–52, 2018.

[29] C. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A python package for symbolic music processing," in *Music Encoding Conference (MEC)*, 2022.

[30] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *Workshop on Representation Learning on Graphs and Manifolds at International Conference on Learning Representations (ICLR)*, 2019.

[31] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *International Conference on Machine Learning*. PMLR, 2017, pp. 3319–3328.

[32] Z. Ying, D. Bourgeois, J. You, M. Zitnik, and J. Leskovec, "Gnnexplainer: Generating explanations for graph neural networks," *Advances in neural information processing systems*, vol. 32, 2019.

[33] M. S. Schlichtkrull, N. De Cao, and I. Titov, "Interpreting graph neural networks for NLP with differentiable edge masking," 2021.

[34] S. Marlowe, "Schenkerian analysis of fugue: A practical demonstration," *Journal of Music Theory Pedagogy*, vol. 33, no. 1, p. 6, 2019.

[35] J. Hentschel, M. Neuwirth, and M. Rohrmeier, "The annotated Mozart sonatas: Score, harmony, and cadence," *Transactions of the International Society for Music Information Retrieval*, vol. 4, no. 1, pp. 67–80, 2021.

[36] J. Swinkin, "Schenkerian analysis, metaphor, and performance," in *College Music Symposium*, vol. 47. JSTOR, 2007, pp. 76–99.

PATTERNS UI, AN INTERACTIVE TOOL FOR MUSIC EXPLORATION

Rory SWEENEY (rory.sweeney@universityofgalway.ie) (0009-0005-5439-2032)[1],
Pushkar JAJORIA (pushkarjajoria@gmail.com) (0009-0006-3789-5372)[1],
Danny DIAMOND (d.diamond1@universityofgalway.ie)[1],
Mathieu D'AQUIN (mathieu.daquin@loria.fr) (0000-0001-7276-4702)[2], and
James MCDERMOTT (james.mcdermott@universityofgalway.ie) (0000-0002-1402-6995)[1]

[1]**Data Science Institute and School of Computer Science, University of Galway**, Galway, Ireland
[2]**Laboratoire Lorrain de Recherche en Informatique et Applications and Institut des Sciences du Digital, Université de Lorraine**, Nancy, France

ABSTRACT

Patterns UI is a novel user interface designed for the exploration of trends within symbolic music corpora through shared melodic patterns. It is an open source tool that caters to musicologists, musicians and enthusiasts, enabling users to discover, compare, and visualize melodic patterns in European folk music. These patterns are represented as short, integer n-grams of accented notes, based on previous musicological research. A central feature is an interactive network visualisation showing tunes, patterns, and the relationships among them, as well as tune families. The interface also features a fuzzy search for tune titles, a search by pattern, and a metadata search. Patterns UI was co-designed and then evaluated with potential users, including musicologists.

1. INTRODUCTION

Patterns UI is a simple, modular piece of software for exploring a collection of tunes and their shared melodic patterns, created using previously-developed pattern detection algorithms, stored as a knowledge graph (KG). In particular it allows direct exploration of the graph as nodes and edges, with two different views, and for various types of search. A main view is shown in Figure 1.

The goal of the work presented was to develop an interactive tool for exploring and visualising the information contained in the KG to satisfy the needs of musicologists, composers, musicians and enthusiasts. As these users might not possess the proficiency to derive meaningful insights from KGs through direct SPARQL queries against the KG, our objective was to design an interface facilitating users to search and explore within the KG, taking advantage of visualisation, search, information hiding, and more.

The data to be explored in the Patterns UI is a database of melodic patterns extracted from Irish folk tunes. The tune corpus is *The Session*, a central resource in the Irish folk community [1]. The familiar n-gram method is used to extract all relevant patterns from a subset of the corpus. An n-gram in our context is a sequence of consecutive notes of length n in a tune. We also extract a measure of their importance in the tune, based on TF-IDF. TF-IDF, or *Term Frequency–Inverse Document Frequency*, is a method of determining the importance of a term, or in our case an n-gram, to a document by comparing its frequency in the document to its occurrences across an entire corpus. An n-gram with a high TF-IDF value is common in the tune but not common in the overall corpus. Using this method, the melodic patterns most characteristic of each tune can be identified. Finally we extract a complexity measure, based simply on the number of unique notes in the n-gram pattern. The patterns are stored in a knowledge graph (KG) together with metadata on tunes.

Since a tune contains multiple patterns, and the same pattern can be contained in multiple tunes, our data forms a network. Moreover, we have metadata on *Tune Family* membership for a small subset of the corpus, which can be explored and visualised in the network. We will describe the Irish traditional music context, the pattern extraction, metadata, and tune family data in more detail in Section 2. This section also includes information on the processing of pattern data to produce a KG.

Our focus is then on how to explore such a KG. Starting from competency questions (CQs) we have drawn inspiration from previous UIs for exploration of musical corpora and of networks in general, described in Section 3. We engaged in a co-design process with potential users such as computational musicologists. We describe this process in Section 4, leading to a statement of requirements.

We describe the app itself in Section 5, with discussion and evaluation of strengths and limitations. Notes on evaluation are in Section 6. We offer conclusions and future work in Section 7.

2. BACKGROUND

2.1 Patterns and Tune Families in Irish Folk Music

The existence of common melodic patterns[1] between tunes in the Irish tradition was first noted by 19th century collectors George Petrie and William Forde [3]. The influential Irish-American music collector Francis O'Neill de-

[1] The following section draws on the M.Sc. thesis of author Danny Diamond [2].

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

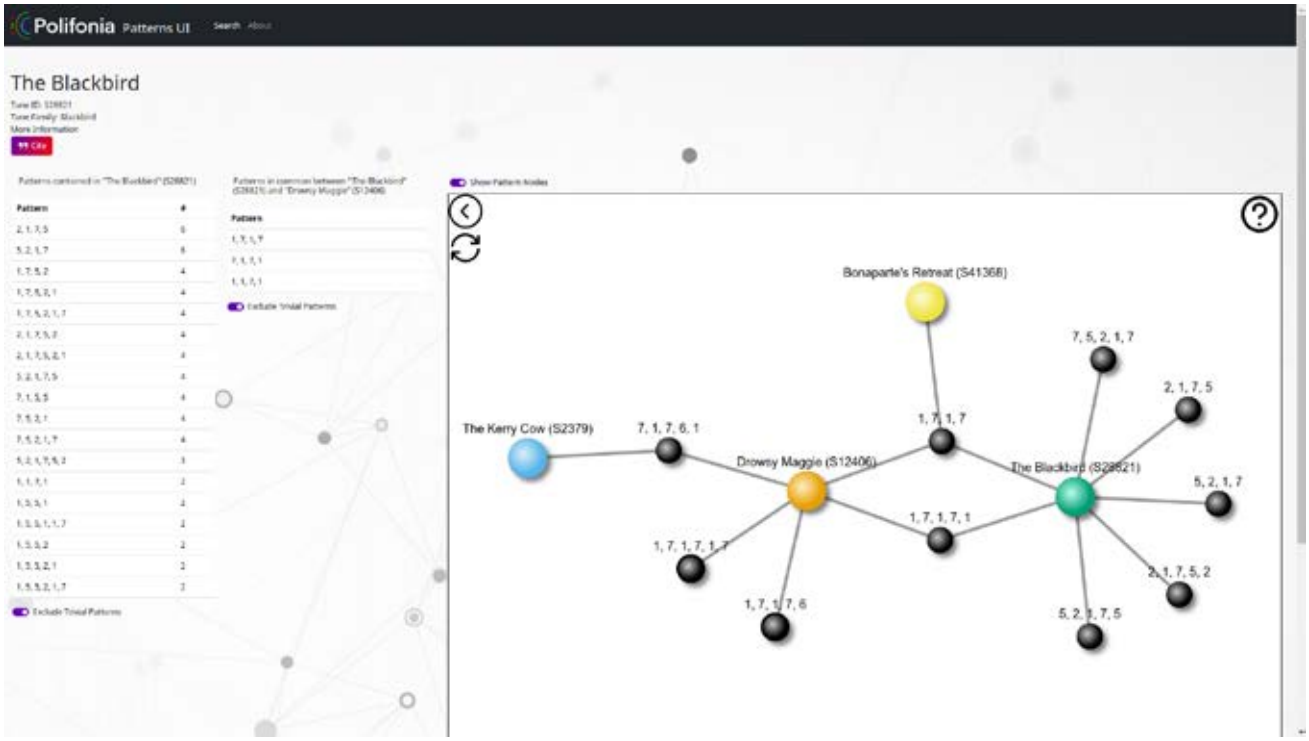


Figure 1. Patterns UI *Composition* page. On the left, a table of the most common patterns in the selected tune; a middle panel containing a list of patterns in common with the previously selected tune; on the right, a network visualisation showing the current tune, some of its most common patterns (small, black nodes), and other tunes that contain the same pattern. At the top of the page, the tune title and tune family name are displayed above a link to the original source for the tune and a *Cite* button.

voted a chapter in his 1913 book Irish Minstrels and Musicians to the qualitative identification of related melodies within the Irish tradition [4].

In 1950, American folk music collector and researcher Samuel Bayard published his theory of *tune families*: proposing that many traditional melodies could be traced, in the manner of a family-tree, to prototype melodies [5]. In Bayard's words:

"*...a group of melodies showing basic interrelation by means of constant melodic correspondence, and presumably owing their mutual likeness to descent from a single air that has assumed multiple forms through processes of variation, imitation and assimilation.*"

Bayard's tune family concept has been widely adopted in research on Irish and other Western folk music [3, 6, 7].

This provides the point of departure for our work. We assume that these important shared patterns can be captured well by a diatonic representation. Indeed, the diatonic representation deals well with the modal ambiguity common in Irish traditional music. We deal with transpositions among tune variants by normalising all melodies to a common tonal centre.

Further, we take advantage of previous work on structure in Irish traditional music to focus on "accented tones" only, i.e. those on the beat. Their importance (in contrast to the variability of the unaccented tones) in defining a stable outline melody crops up consistently in the literature [6,8–10].

Thus, while we have worked with multiple definitions of patterns, in the current work we represent a pattern as an n-gram of integers representing a sequence of diatonic scale degrees of accented notes, which occurs more than once in the corpus. For example, the pattern (7, 1, 7, 6, 1) is a 5-gram pattern visible in Fig. 1.

The Irish folk music corpus "The Session" [2] (approximately 40,000 tunes) was used as a source of tune data in ABC notation, and converted to MIDI and then to the integer representation. For 314 tunes, we have annotated them with ground-truth on tune family membership. A further source is the *Essen* corpus [3], with tune data available through the MTCFeatures library [4].

2.2 Linked Data

Throughout the Polifonia project, we have represented datasets as linked data (ontologies and KGs). The motivation for this approach was to enable flexible and efficient querying, open access, preservation, and integration across multiple data sources [11]. Using well-designed ontologies and KGs allows current data to integrate well with future data also.

So, the patterns we have extracted are stored in a Patterns KG, which itself takes advantage of ("lives on top of") an additional KG representing tune metadata [5]. The construction of these KGs is described in previous work [11],

[2] https://thesession.org/
[3] http://essen.themefinder.org/
[4] https://github.com/pvankranenburg/MTCFeatures
[5] https://github.com/polifonia-project/tunes-knowledge-graph

and draws on previous work in ontologies and KG creation within the Polifonia project [12–14].

The *Patterns2KG* pipeline was used to extract patterns from the tune data and process them into a KG in RDF format, together with tune family membership data. Information is obtained from the KG using the SPARQL query language [11], typically via a SPARQL endpoint.

3. PREVIOUS WORK

Tovstogan et al. used a user interface for the exploration and discovery of personal music collections in which labelled segments of musical compositions are represented as points in a 2D visualisation. Dimension reduction algorithms were used to position points in the visualisations and suggest similarity. They found that users perceived this system as engaging, rewarding and useful. The authors suggest that similar systems could be extended to larger music collections [15].

Knees et al. developed a three-dimensional interface for exploration of musical collections, in which similar compositions are clustered together, and their properties processed into a height value to form a landscape through which the user can navigate a music collection. A user study of 8 people received positive responses [16].

The *Digital Music Lab* system of Abdallah et al. employs semantic web technologies to support large-scale musicological research across large music corpora. Their system integrates into Linked Open Data, allowing for distributed system across multiple corpora. It includes an interactive graphical interface through which analyses of musical collections can be visualised as bar charts, line graphs or histograms. Their user interface does not, however, include a network visualisation. [17].

Walshaw developed the abcnotation.com web interface to search an online corpus of traditional music stored in ABC notation. The interface provides tools to view tunes in musical notation and play MIDI audio. The interface also features the *TuneGraph* network visualisation, which presents tunes as nodes with connections to a small set of similar tunes. Nodes can be double-clicked to navigate to the selected node's page, with a new small network diagram generated for that tune. The *abcnotation.com* interface is depicted in Figure 2 [18–20].

The *TunePal* interface, `https://tunepal.org`, allows querying a corpus of traditional Irish, Welsh, Scottish and Breton music by title and by playing music. The score of the selected composition is displayed on a musical staff and in ABC notation. MIDI and recorded audio of tunes can be played. There are no visualisations on the *TunePal* webpage, and the target audience is musicians rather than musicologists. [21–23]

As part of the Polifonia project, a harmonic relation graph was produced, showing inferred relationships between pairs of pieces of music (see Figure 3). This was presented at the *Sonar Festival* Music and AI event in October 2021. In this graph, each edge represents an inferred relationship between a pair of pieces of music. This graph is static, but uses graph layout algorithms to produce a strong sense of overall geometry of the implicit 2D musical space.

Figure 2. The *Abcnotation* interface, e.g. `https://abcnotation.com/tunePage?a=tunearch.org/wiki/Walnut_Gap.no-ext/0001` demonstrates the visualisation of a small network of similar tunes.

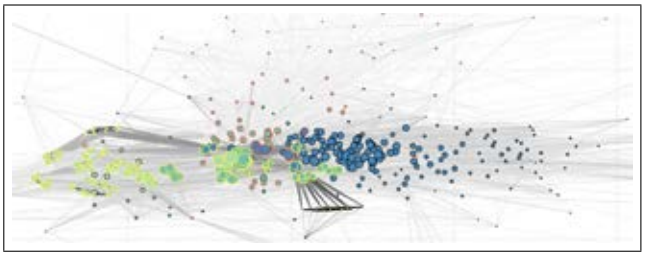


Figure 3. A harmonic relation graph, part of the Polifonia INTERLINK pilot. Figure from [24].

Finally, we mention the *LOD Live* interface, designed for exploration of arbitrary KGs, with an example shown in Figure 4. This UI is not music-oriented, but served as inspiration for a live network diagram with show/hide functionality based on linked open data.

This review of pre-existing interfaces produced a list of desirable interface features including search facilities over metadata and over contents, display of patterns on musical staves, audio playback of patterns and a network diagram for exploration of relationships.

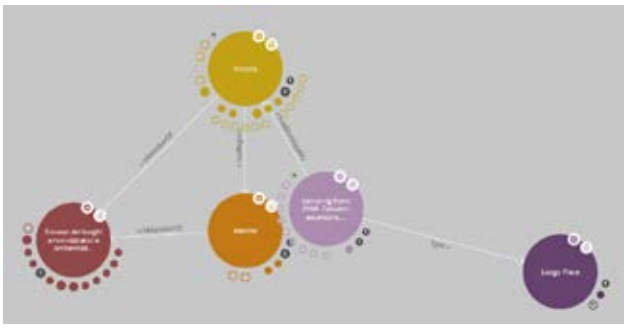


Figure 4. The *LOD Live* interface, e.g. `http://en.lodlive.it/?https://w3id.org/italia/env/ld/place/municipality/00201_042002`.

4. DESIGN PROCESS

This section describes how we went about designing the interface and software, incorporating potential users' feedback at early stages.

4.1 Wireframe Design

The initial phase of our project involved the task of designing the website's wireframe. This step helped to decide the overall structure of the website, ensuring that it was both user-friendly and tailored to the specific needs of our target audience – musicologists. The design phase also involved planning of key features, such as a search interface that borrowed elements from previous implementations while remaining flexible to accommodate various user input formats. Wireframe designs for the *Search* interface, *Composition* page and *Pattern* page were particularly used during our co-design process.

4.2 Co-Design Process

A significant aspect of our Co-Design process was conducting remote, semi-structured interviews with musicologists. Initial wireframes and/or related interfaces were presented to each participant, followed by an open discussion with brainstorming about possible features. Sessions lasted 20–50 minutes. Six such interviews were conducted with musicologists from varying backgrounds.

The insights provided by these discussions into the practical applications and user expectations of our website were instrumental in designing our user interface. The proposal of a network diagram as a method of interaction was received particularly positively, underlining its potential as a musicological tool. The minutes and feedback from these meetings were archived and guided the work that followed.

Key takeaways included the need for a search interface leading to a list of results, the importance of visual cues like hovering and highlighting for information presentation, and the necessity of a flexible network diagram to explore and navigate the large corpus. Furthermore, these interactions emphasized the importance of catering to both casual and expert users, balancing immediate usability with the depth of information for more experienced musicologists.

During the design phase, the need for citeability of particular findings was also raised by project data management researchers.

4.3 Stories and Competency Questions

Within the Polifonia project, resulting from stakeholder interviews, personas and competency questions (CQs) were created to guide design processes within the project. We drew on these during design and development [6] . The ones most relevant to our work are listed below.

For the persona *Brendan*, who is a traditional musician and a lecturer in ethnomusicology and Irish traditional music, we sought to address the following CQs.

- What tunes are similar to tune X, given similarity measure Y?
- What are the metadata for collection X?

Persona *Mark* is a computational musicologist, researcher and lecturer. The following CQs were created to address his needs from the system.

- What are all compositions in tune family X?
- What are the similarities / differences of all compositions in tune family X according to measure Y?

The persona *Sophia* is a musicologist and a practising musician, for whom the following CQ was considered.

- What different motifs exist in a piece of music?

4.4 Requirements

Based on the above, we arrived at the following list of requirements:

- An *About* page with information, links, and explanations.
- A *Search* function, allowing search by different fields, including text fields like 'title', and structured fields like 'key signature'.
- Fuzzy *Title* search to help deal with variant spellings of titles.
- Search by pattern, allowing different methods of representing a pattern (e.g., space-separated integers versus comma-separated).
- A *Composition* page, showing details of a particular tune, relevant patterns, and external links to additional information.
- A *Pattern* page, to show the details and relevant tunes for a particular pattern, and visualise a pattern on a musical staff, with the ability to listen to it directly.
- A network diagram, showing tunes and patterns as nodes, with links representing a pattern present in a tune.
- A feature to indicate tune family membership.
- A feature to allow citation of any finding, e.g. by citing a specific URL.

4.5 Technology Stack

We created a modular design. The KGs, developed previously, are hosted in a Blazegraph instance via a Polifonia-custom Flask app providing a SPARQL endpoint [7] .

The backend server was written using the Python Flask web framework, and communicates with the KG using the SPARQL query language. The Vue.js JavaScript framework was used for front-end development.

The source code repositories for both the frontend [8] and backend [9] are open and accessible on GitHub.

[6] https://github.com/polifonia-project/stories

[7] https://polifonia.disi.unibo.it/fonn/sparql

[8] https://github.com/polifonia-project/pattern-exploration-GUI

[9] https://github.com/polifonia-project/pattern-explorations-backend

5. USER INTERFACE

In this section, the final design of the application is described from the point of view of the user. The interface opens onto a *Search* page which features three search methods for finding tunes.

The first search method is a tune *Title* search. The fuzziness of its matching algorithm allows it to accept partial titles or misspellings as search terms and still return relevant results. The search results show the tune name and tune ID for each result, as well as relevant metadata such as tune type (e.g. jig, reel, strathspey), key and time signature, as shown in Figure 5. Clicking on a search result opens the *Composition* page for the selected tune.

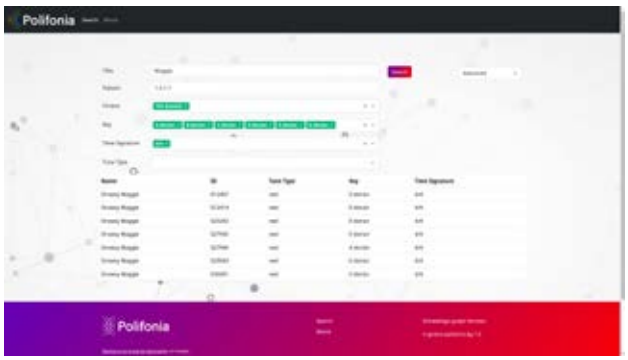


Figure 5. The *Search* page showing the *Advanced* search interface. The *Title* field contains the partial title 'Maggie', which can match a number of tune titles as a result of the fuzzy *Title* search feature. The pattern '1, 3, 1, 7' has been entered in the *Pattern* field using hyphens as a delimiter. 'The Session' has been selected from the *Corpus* drop-down. The *Key* drop-down further limits the search to tunes featuring a Dorian key, and the *Time Signature* drop-down specifies a '4/4' time signature. The results of the search are shown in a table.

The second type of search is a *Pattern* search, which finds tunes containing a melodic pattern inputted as a sequence of delimited numeric scale degrees.

Finally, the *Advanced* search incorporates the fuzzy *Title* and *Pattern* search fields, along with multiple-select drop-down boxes for 'Corpus', 'Key', 'Time Signature', and 'Tune Type', the options of which can be filtered using associated text fields.

An example of a *Composition* page is shown in Figure 1. On the *Composition* page, the most prominent feature is a network diagram. This represents tunes as large nodes, coloured according to the tune's family. The slightly smaller *Pattern* nodes, in black, represent melodic patterns. A connection between a *Tune* node and a pattern node indicates that that pattern is contained in the tune in question.

The network visualisation is interactive and can be expanded by clicking on nodes. A *Tune* node can be clicked to reveal additional patterns contained in the selected tune, and a *Pattern* node can be clicked to reveal additional tunes containing that pattern. Double-clicking a tune node navigates to the clicked node's *Composition* page, while double-clicking a *Pattern* node navigates to this node's *Pattern* page. If there are no additional connected nodes to expand when a node is clicked, the perimeter of the clicked node flashes red to indicate this.

Nodes in the network diagram can be rearranged by dragging with the mouse pointer, and the diagram can be zoomed by scrolling. Two icons at the top left of the diagram allow for undo and reset functions for the visualisation. A *Help* icon in the top-right corner of the visualisation, when clicked, displays a short series of tutorial panes that explain how to use the network visualisation. An example of one of these tutorial panes is shown in Figure 6.

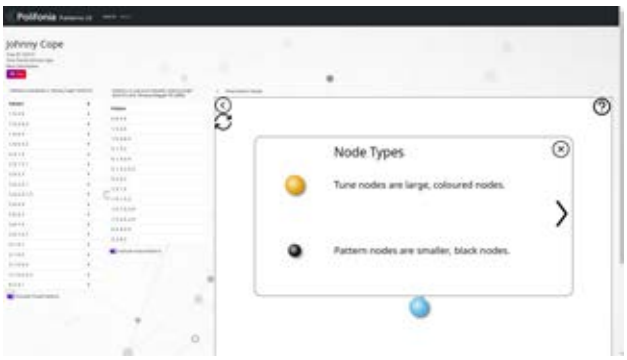


Figure 6. The network visualisation tutorial pane is opened by clicking on the *Help* icon. The tutorial panel shown lists the types of nodes used in the visualisations.

A toggle switch above the network visualisation allows the pattern nodes to be hidden, resulting in a network visualisation containing only tune nodes connected to other tune nodes based on shared patterns. This configuration is shown in Figure 7.

Figure 7. The *Composition* page featuring the network diagram with pattern nodes disabled. This network diagram shows the selected tune and similar tunes based on the numbers of occurrences of shared patterns weighted by pattern complexity. Contrast this with Figure 9 which shows the same tunes, but also shows the patterns.

Additionally, the *Composition* page features a panel on the left side listing the most common melodic patterns contained in the selected tune along with their occurrence frequencies. Trivial patterns can be excluded from the list by adjusting a toggle switch.

Trivial patterns are patterns with low pattern complexity. Pattern complexity is a measure of the importance of a pattern in a tune, defined as the proportion of unique note values in the pattern. For example, the pattern "5, 5, 5, 5" is very common in the corpus. However, since it is repetitive, its pattern complexity is low, and so it is likely not a distinctive melody associated with that tune. It is de-emphasised as a result of its low complexity.

When a tune node is double-clicked, the selected tune's *Composition* page will load and feature a middle panel which lists the melodic patterns in common between this tune and the tune described by the previous *Composition* page. Here too, trivial patterns can be excluded by adjusting a toggle switch. The patterns listed in both panels can be clicked to navigate to their respective *Pattern* pages.

Figure 8. The *Pattern* page shows a musical stave featuring the pattern in musical notation, as well as a MIDI player which can play audio of the pattern. Below them is the full list of tunes containing the selected pattern. Below the page title, is the *Cite* button.

The *Pattern* page features a musical stave representing the melodic pattern described by the page in musical notation, along with a MIDI player that can play audio of the pattern. Below the stave, there is a list of the tunes that contain the selected pattern, each entry linking to its respective *Tune* page. A screenshot of the *Pattern* page can be seen in Figure 8.

Figure 9. The *Tune Family* page shows a list of tunes contained in the selected tune family. Below the tune family page title, there is the *Cite* button.

The *Tune Family* page for a given tune can be accessed from a link on its *Composition* page below the page title. The *Tune Family* page lists all the tunes in the selected tune family. Each tune in the list links to its respective *Composition* page. An example of a *Tune Family* page is shown in Figure 9. The *Composition* page also features a link to external resources for each tune.

The *About* page provides information about the interface, including links to the project GitHub repositories, information about the Polifonia project, the funding associated with the project and a link to a page providing the Patterns knowledge graph endpoint and example SPARQL queries that can be used with it.

On several pages, there is a feature that generates a citation for the *current* page, which can be copied to the clipboard using the *Copy* button. This provides easy citability for specific findings, such as a specific pattern's presence in a specific tune.

The footer on each page displays the current *Patterns* knowledge graph version string as well as links to the *Polifona Web Portal* and Polifonia social media pages.

6. EVALUATION

6.1 Usability Testing

Usability testing evaluations were conducted with 2 participants (not part of the project) during February 2024.

- Subject 1 is a PhD student, working on musical complexity and patterns, and music generation, working on both Irish and Iranian melodic music.
- Subject 2 is an experienced music technology researcher, working as a lecturer.

For each evaluation, the subject was given a very brief explanation of the UI and KGs - tunes, patterns, links between them, and searches. The subject was then given several tasks to be carried out, while two observers watched, made notes, answered Subject's questions, and gave hints, only when needed. Sessions lasted approximately 30 minutes. The tasks given to the subjects were derived from our co-design process, competency questions, and requirements, as described in previous sections.

Subjects were generally successful in tasks involving searches. The fuzziness of the title search proved useful for unfamiliar song names, and participants had no difficulty with pattern searches. For advanced searches, Subject 1 did not initially realise that multiple drop-down options could be selected, but both participants eventually completed the task successfully.

The purposes of the pattern panes were quite clear to both participants, though the appearance of the middle pane, for patterns in common with the previous tune, was not initially noticed by Subject 2.

Some features of the network visualisation were confusing for the participants. The existence of the hover feature to reveal the tune family name was not initially apparent to Subject 1, and the existence of distinct single and double click features was not clear. The red ring flash visual feedback for the case when no additional neighbour nodes can be expanded was not noticed by one participant.

Despite this, both participants were able to successfully achieve tasks related to the network visualisation such as

finding additional tunes containing a given pattern and finding patterns in common between two tunes.

Some of the links on the composition page and in the page footer were found not to be sufficiently prominent and not found by the participants.

Using the *Pattern* page to play MIDI audio of patterns and using the *Tune Family* page to find tunes in the same family as the selected tune were easily accomplished by both participants.

6.2 Walkthrough Demonstration and Evaluation

An open walkthrough and demonstration of the interface was held with a large group of project partners during February 2024. The participants included academics in the fields of Computer Science, Human-Computer Interaction and Musicology. This involved a ten-minute, remote demonstration of all features of the application, in combination with the option of participants using the interface directly through a provided link to the live website. The most important feedback here was to improve user-friendliness for first-time users by providing better visual differentiation between elements of the network diagrams, and pop-up help; and to provide code resources such as SPARQL queries on a Resources page.

All feedback was acted on for the current version of the UI described in the previous section.

7. CONCLUSIONS AND FUTURE WORK

In this paper we have presented a novel user interface, Patterns UI, for exploration of a collection of tunes and patterns. Usability testing produced positive reactions, demonstrated the interface's usability, and produced useful feedback. The UI clearly answers the competency questions.

Planned future work on the system includes offering multiple tune similarity metrics with which to link tunes in the tune-tune network visualisation, such as alignment similarity and similar methods developed by Van Kranenburg and colleagues [25, 26].

In contrast to our melody-focussed work, other methods of measuring relatedness between pairs of pieces of music could also be used. We plan to use the UI for interactive visualisation of relationships among pieces of music based on chord- and harmony-based knowledge graphs, such as the *ChoCo* or *Harmory* KGs [27–29].

Acknowledgments

Patterns UI, as part of the Polifonia Project, has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement N. 101004746. This publication has emanated from research supported in part by a grant from Science Foundation Ireland under Grant number [12/RC/2289_P2] the Insight SFI Research Centre for Data Analytics. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

8. REFERENCES

[1] (2024) The Session. Accessed 2024-02-07. [Online]. Available: https://thesession.org/

[2] D. Diamond, "Automatic tune family detection in a corpus of Irish traditional dance tunes," Master's thesis, University of Galway, 2024, forthcoming.

[3] J. R. Cowdery, "A fresh look at the concept of tune family," *Ethnomusicology*, vol. 28, no. 3, p. 495, Sep. 1984. [Online]. Available: https://www.jstor.org/stable/851236?origin=crossref

[4] F. O'Neill, *Irish minstrels and musicians*. Chicago, USA: Regan Printing House, 1913.

[5] S. P. Bayard, "Prolegomena to a study of the principal melodic families of British-American folk song," *The Journal of American Folklore*, vol. 63, no. 247, pp. 1–44, 1950, publisher: University of Illinois Press. [Online]. Available: http://www.jstor.org/stable/537347

[6] M. Ó Suilleabháin, "The creative process in Irish traditional music," in *Irish musical studies 1: musicology in Ireland*, ser. Irish musical studies. Dublin, Ireland: Irish Academic Press, 1990, vol. 1, pp. 117–130.

[7] A. Hillhouse, "Tradition and innovation in Irish instrumental folk music," Master's thesis, The University of British Columbia,, Vancouver, Canada, 2005.

[8] B. Breathnach, "Between the jigs and the reels," 1982. [Online]. Available: http://msikio.online.fr/Breathnach/breandn.htm

[9] D. Ó Maidín, "A programmer's environment for music analysis," Ph.D. dissertation, University of Limerick, Limerick, Ireland, 1995. [Online]. Available: http://rgdoi.net/10.13140/2.1.3387.2969

[10] C. Walshaw, "A multilevel melodic similarity framework," in *Extended Abstracts for the Late-Breaking Demo Session of the 16th International Society for Music Information Retrieval Conference, 2015*, Málaga, Spain, 2015.

[11] A. Shahid, D. Diamond, and J. McDermott, "Patterns2KG: JAMS pipeline for modeling music patterns," in *International Joint Workshop on Semantic Web and Ontology Design for Cultural Heritage*, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:266377142

[12] J. de Berardinis, V. A. Carriero, N. Jain, N. Lazzari, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "The Polifonia Ontology Network: Building a semantic backbone for musical heritage," in *International Semantic Web Conference*. Springer, 2023, pp. 302–322.

[13] V. Carriero, J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "The Music Meta Ontology: A flexible semantic model for the interoperability of music metadata," in *ISMIR 2023 Hybrid Conference*, 2023.

[14] J. De Berardinis, A. Meroño Peñuela, A. Poltronieri, and Presutti, "The Music Annotation Pattern," in *WOP2022: 13th Workshop on Ontology Design and Patterns*, Hangzhou, China, 2022.

[15] P. Tovstogan, X. Serra, and D. Bogdanov, "Visualization of deep audio embeddings for music exploration and rediscovery," in *Proceedings of the 19th Sound and Music Computing Conference*, 2022, pp. 493–500.

[16] P. Knees, M. Schedl, T. Pohle, and G. Widmer, "An innovative three-dimensional user interface for exploring music collections enriched," in *Proceedings of the 14th ACM International Conference on Multimedia*, ser. MM '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 17–24. [Online]. Available: https://doi.org/10.1145/1180639.1180652

[17] S. Abdallah, E. Benetos, N. Gold, S. Hargreaves, T. Weyde, and D. Wolff, "The digital music lab: A big data infrastructure for digital musicology," *J. Comput. Cult. Herit.*, vol. 10, no. 1, jan 2017. [Online]. Available: https://doi.org/10.1145/2983918

[18] C. Walshaw, "A visual exploration of melodic relationships within traditional music collections," in *2018 22nd International Conference Information Visualisation (IV)*, 2018, pp. 478–483.

[19] ——, "Constructing proximity graphs to explore similarities in large-scale melodic datasets." in *6th Intl Workshop on Folk Music Analysis*, 2016, pp. 22–29.

[20] ——, "TuneGraph, an online visual tool for exploring melodic similarity," in *Proc. Digital Research in the Humanities and Arts*, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:2605457

[21] B. Duggan and B. O'Shea, "Tunepal - disseminating a music information retrieval system to the traditional Irish music community," in *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010*, J. S. Downie and R. C. Veltkamp, Eds. International Society for Music Information Retrieval, 2010, pp. 583–588. [Online]. Available: http://ismir2010.ismir.net/proceedings/ismir2010-100.pdf

[22] B. Duggan, "Tunepal: the traditional musician's toolbox," in *Proceedings of the Second Workshop on EHeritage and Digital Art Preservation*, ser. eHeritage '10. New York, NY, USA: Association for Computing Machinery, 2010, p. 25–30. [Online]. Available: https://doi.org/10.1145/1877922.1877931

[23] B. Duggan and B. O'Shea, "Tunepal: Searching a digital library of traditional music scores." *OCLC Systems & Services*, vol. 27, 10 2011.

[24] J. McDermott, D. Diamond, A. Shahid, A. Poltronieri, A. Meroño Peñuela, R. Fournier-S'niehotta, and T. Zhu, "Polifonia deliverable d3.2: Analysis of music repositories to identify musical patterns," Polifonia Consortium, Tech. Rep., 2021, https://polifonia-project.eu/dissemination/.

[25] A. Volk and P. Van Kranenburg, "Melodic similarity among folk songs: An annotation study on similarity-based categorization in music," *Musicae Scientiae*, vol. 16, 2012.

[26] B. Janssen, P. van Kranenburg, and A. Volk, "Finding Occurrences of Melodic Segments in Folk Songs Employing Symbolic Similarity Measures," *Journal of New Music Research*, vol. 46, no. 2, pp. 118–134, Apr. 2017. [Online]. Available: https://www.tandfonline.com/doi/full/10.1080/09298215.2017.1316292

[27] J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "ChoCo: a chord corpus and a data transformation workflow for musical harmony knowledge graphs," *Scientific Data*, vol. 10, 09 2023.

[28] J. de Berardinis, A. Meroño Peñuela, A. Poltronieri, and V. Presutti, "The Harmonic Memory: a knowledge graph of harmonic patterns as a trustworthy framework for computational creativity," in *Proceedings of the ACM Web Conference 2023*, ser. WWW '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 3873–3882. [Online]. Available: https://doi.org/10.1145/3543507.3587428

[29] N. Lazzari, A. Poltronieri, and V. Presutti, "Pitch-class2vec: Symbolic music structure segmentation with chord embeddings," in *Proceedings of the 1st Workshop on Artificial Intelligence and Creativity co-located with 21th International Conference of the Italian Association for Artificial Intelligence (AIxIA 2022)*, 2022, pp. 14–30. [Online]. Available: https://api.semanticscholar.org/CorpusID:254223385

PAPER & POSTER /DEMO SESSION 2

/ EXPLORING SAMPLING STRATEGIES IN LATENT SPACES FOR MUSIC GENERATION

/ GUITAR CHORD DIAGRAM SUGGESTION FOR WESTERN POPULAR MUSIC

/ A SONIFICATION METHOD FOR MONITORING CHEMICAL SENSOR DATA

/ INTERACTIVE MUSICAL PERIODIC TABLE: SONIFICATION OF VISIBLE ELEMENT EMISSION SPECTRA

/ INTERPRETATIVE DATA SONIFICATION: USING LLMS TO INTERPRET DATA AND GENERATE CONTINUOUS SOUNDSCAPES AT THE SYDNEY OPERA HOUSE

/ ARPEGGIATORUM: AN AUDIO CONTROLLABLE MIDI ARPEGGIATOR

/ SOUNDSCAPE PERSONALISATION AT WORK: DESIGNING AI-ENABLED SOUND TECHNOLOGIES FOR THE WORKPLACE

/ A GENERATIVE FRAMEWORK FOR COMPOSITION-AWARE LOOP RECOMMENDATION IN MUSIC PRODUCTION: DRUM2BASS USE CASE

EXPLORING SAMPLING STRATEGIES IN LATENT SPACES FOR MUSIC GENERATION

Nádia CARVALHO (nscarvalho@fe.up.pt) (0000-0001-6882-5138)[1] and
Gilberto BERNARDES (gba@fe.up.pt) (0000-0003-3884-2687)[1]

[1]**University of Porto – Faculty of Engineering and INESC TEC**, Porto, Portugal

ABSTRACT

This paper investigates sampling strategies within latent spaces for music generation, focusing on (chordified) J.S. Bach Chorales and utilizing MusicVAE as the generative model. We conduct an experiment comparing three sampling and interpolation strategies within the latent space to generate chord progressions – from a discrete vocabulary of Bach's chords – to Bach's original chord sequences. Given a three-chord sequence from an original Bach chorale, we assess sampling strategies for replacing the middle chord. In detail, we adopt the following sampling strategies: (1) traditional linear interpolation, (2) k-nearest neighbors, and (3) k-nearest neighbors combined with angular alignment. The study evaluates their alignment with music theory principles of functional harmony embedding and voice-leading to mirror Bach's original chord sequences. Preliminary findings suggest that k-nearest neighbors and k-nearest neighbors combined with angular alignment closely align with the tonal function of the original chord, with k-nearest neighbors excelling in bass line interpolation and the combined strategy potentially enhancing voice-leading in upper voices. Linear interpolation maintains aspects of voice-leading but confines selections within defined tonal spaces, reflecting the non-linear characteristics of the original sequences. Our study contributes to the dynamics of latent space sampling for music generation, offering potential avenues for enhancing explainable creative strategies.

1. INTRODUCTION

The rise of generative models in music composition has sparked a notable interest in delving into the latent spaces of musical data for creative pursuits [1–3]. As AI-driven technologies advance, there is a growing curiosity about the intricate mechanisms underlying these models and their potential to unlock new realms of musical expression. Latent spaces, in particular, present an enticing prospect for composers and researchers, offering a rich landscape where musical ideas can be explored, manipulated, and synthesized in innovative ways [4–6]. This fascination has led to investigations into the dynamics of latent space navigation, aiming to unveil the hidden structures governing musical composition and interpretation [5, 7–9].

The pursuit of uncovering meaningful music semantics within latent spaces for controllable sampling based on distinct music characteristics is a vibrant area of investigation within the realms of music informatics and artificial intelligence [9–12]. This research endeavor is driven by the overarching goal of equipping users, musicians, and composers with sophisticated tools, enabling them to engage in interactive exploration and creation of music tailored to their preferences and specifications. The potential applications of such advancements span a broad spectrum, encompassing domains like music composition, production, interactive entertainment, and personalized music recommendation systems [9].

Exploring a latent space involves navigating its multi-dimensional structure to capture the essence of music semantics. This capability empowers users to transition between different representations, each embodying a unique combination of musical traits, facilitating the exploration of a multitude of musical variations and possibilities. Additionally, interpolation, a traditional method of exploring latent spaces, involves seamlessly transitioning between established music representations within the latent space, thereby facilitating smooth transitions and novel combinations [13, 14].

In this context, our study delves into the realm of music generation, seeking to elucidate the role of sampling strategies within latent spaces, with a particular focus on J.S. Bach Chorales as a benchmark test set. Our primary objective is to evaluate the efficacy of different sampling and interpolation strategies within the latent space, using MusicVAE [15] as our generative model.

The fundamental aim of our evaluation is to ascertain the extent to which these strategies can accurately generate chord progressions that adhere to Bach's original sequences. To achieve this, we conduct an experiment comparing three distinct sampling strategies. These strategies involve departing from a chorale's chord progression and exploring chord substitutions based on the previous and sequential chord contexts. For instance, given a chord sequence [A, B, C], our goal is to generate new sequences such as [A, B', C], [A, B'', C], and so forth.

The three sampling strategies under scrutiny include the traditional linear interpolation between adjacent chords, k-nearest neighbors, and a novel approach combining k-nearest neighbors with angular alignment to $\vec{AB}$. By adopting these strategies, we aim to assess their efficacy

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

in preserving the essence of Bach's style while introducing variations that retain semantic coherence within the latent space.

Through our investigation, we aim to provide insights into manipulating latent spaces for music generation, enhancing creative autonomy and stylistic fidelity in algorithmic composition by examining different sampling strategies. Our methodology integrates music theory to assess the similarity between generated sequences and original Bach chorales, allowing us to evaluate stylistic fidelity and delve into latent space semantics. Ultimately, our study seeks to contribute to the ongoing discourse surrounding the intersection of AI and music, paving the way for more sophisticated and nuanced approaches to automated composition.

Our paper is structured as follows. In Section 2, we summarize the sampling strategies employed, namely: (1) traditional linear interpolation between adjacent chords, (2) k-nearest neighbors, and (3) k-nearest neighbors combined with angular alignment to $\vec{AB}$. Section 3 outlines the methodology employed to assess the effectiveness of the sampling strategies in generating chord progressions adhering to a specific musical style. This section provides comprehensive details regarding the experimental setup, including the dataset and the model utilized for the evaluation. Subsequently, Section 4 presents and discusses the obtained results. Finally, Section 5 summarizes the conclusions drawn from our study and suggests potential avenues for future research.

2. SAMPLING STRATEGIES

In this section, we examine sampling techniques enabling us to traverse a discrete set of chords depicted in the latent space and produce novel sequences from a specified chord progression [A, B, C], yielding variations like [A, B', C], [A, B", C], and so on. The pursued strategies include: (1) conventional linear interpolation between consecutive chords, (2) k-nearest neighbors, and (3) a novel approach integrating k-nearest neighbors with angular alignment to $\vec{AB}$.

The subsequent sections delve into each of these strategies in intricate detail, elucidating their principles, methodologies, and implications in the context of our exploration of latent space for music generation.

2.1 Linear Interpolation

Conventional generative music methodologies using VAE latent spaces [13, 14, 16] often entail sampling via linearly interpolated coordinates between two designated musical elements. Musical elements, such as chords, measures, or phrases, can adopt different time scales. Typically, a predetermined number of points are sampled at equal intervals between the starting and ending points, such that:

$$\vec{p} = (1-t)\cdot\vec{p}_0 + t\cdot\vec{p}_1 \quad , \tag{1}$$

where $\vec{p}$ is the interpolated point, $\vec{p}_0$ and $\vec{p}_1$ are the two points between which interpolation is being performed, and t is the interpolation parameter ranging from 0 to 1, determining the weight of $\vec{p}_1$.

This methodology aims to facilitate smooth transitions between predefined musical sequences, predicted on the assumption of a continuous musical spectrum [17]. We sample the dataset to pinpoint the chord nearest to the equally spaced interpolated coordinate to align with our dictionary-constrained chord space.

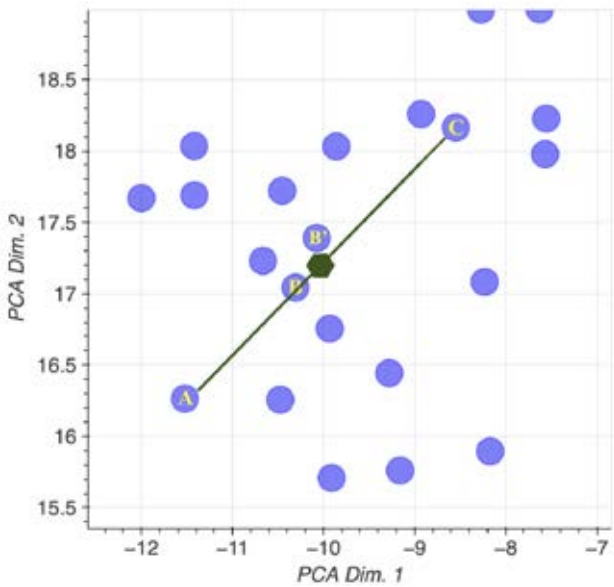


Figure 1. Sampling a chord B' from the latent space (represented in a two-dimensional space) using traditional Linear Interpolation from a sequence of selected chords [A, B, C]. Blue circles represent discrete chords projected in the latent space. The midpoint, found via linear interpolation between A and C, is shown as a dark-green hexagon.

Figure 1 depicts the interpolated sampling technique. Starting from points A and C within the latent space, we interpolate a single point by interpolating linearly between A and C, with a t halfway between 0 and 1. The blue circles in the illustration denote discrete chords projected in the latent space. The midpoint, determined through linear interpolation between A and C, is represented by a dark-green hexagon. In this case, the nearest discrete point is, coincidentally, the original chord B. Thus, the sampled chord, labeled B' in Fig. 1, corresponds to the second nearest discrete chord point.

2.2 k-Nearest Neighbors

The k-nearest neighbors (k-NN) algorithm is a versatile approach commonly employed in classification and regression tasks within supervised learning contexts [18]. This algorithm functions by determining the value of a data point based on the average value of its k nearest neighbors in the feature space. This intuitive principle allows k-NN to adapt effectively to diverse datasets [19].

In this approach, we rely on the perceived significance of the latent space to identify chords that are similar to a given target chord based on their proximity in the embedding space. To sample a chord from the corpus chords

represented in the latent space, we utilize k-NN to choose the k nearest neighbors of a chord within a predefined embedding space. This process, illustrated in Figure 2, involves computing distances (e.g., Cosine or Euclidean) between the target chord and all other chords in the embedding space and selecting the k nearest neighbors. These nearest neighbors serve as candidates for sampling, providing a diverse set of potential chords that are expected to be related to the target chord.

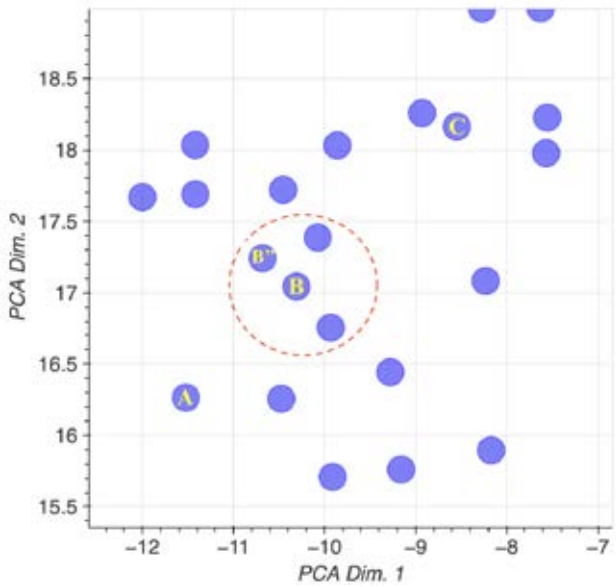


Figure 2. Sampling a chord B" from the latent space (represented in a two-dimensional space) using k-NN. Blue circles represent discrete chords projected in the latent space. The k nearest neighbors (in this case, $k = 3$) are identified within the dotted red line.

In this case, as displayed in Figure 2, we only take B into consideration to find its substitute chord. We search within the nearest neighbors of B (circles within the dotted red line in the Figure; in this case, we use $k = 3$). The closest candidate, according to a measure of distance (e.g., Euclidean or cosine), is notated as B".

2.3 k-Nearest Neighbors and Angular Displacement

We depart from our previous strategy by suggesting a fusion of k-NN with angular alignment, meticulously crafted for the chord pair vector $\vec{AB}$. Angular displacement, used frequently to improve the handling of high-dimensional data whose direction is important [20], involves aligning vectors in a multi-dimensional space according to their angles. Through employing angular alignment to the original chord sequence, our aim is to preserve the overarching direction of the initial sequence.

The initial phase of this sampling strategy mirrors the previous approach: we examine the nearest neighbors of B and curate a subset of the k closest ones using a designated distance metric such as Cosine or Euclidean distance. In this instance, k must encompass a broader selection of chords since we exclusively apply angular alignment to this subset. To compute the angular displacement relative to the vector $\vec{AB}$, we first determine the vector itself. Next, we compute the vector from A to each potential chord point within the set of the k closest ones. Subsequently, we assess the angular disparity between each of these vectors and the original one. The chord displaying the least angular deviation is identified as the optimal replacement.

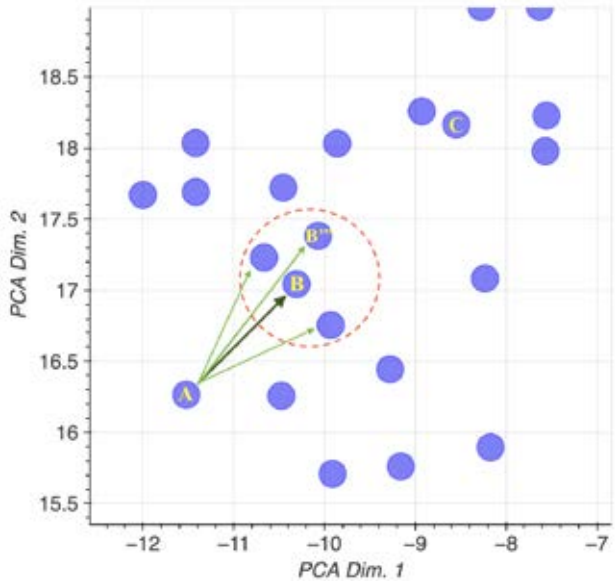


Figure 3. Sampling a chord B''' from the latent space (represented in a two-dimensional space) using k-NN with Angular Alignment to the vector formed from the latent space representation of the selected sequence of chords [A, B]. In this visualization, discrete chords are depicted as blue circles. Within the dotted red line, the k nearest neighbors (in this instance, $k = 3$) are identified. A dark-green arrow indicates the vector from chord A to B, while light-green arrows represent the vectors from chord A to each candidate point.

In Figure 3, we showcase an example wherein we identify the trio of closest neighbors of B and calculate the angular separation between vector $\vec{AB}$ and the extension from A to each prospective point. Even though it isn't the closest neighbor, the closest candidate chord, ensuring optimal continuity, would be the one marked as B''' in the illustration due to its minimal angle.

3. EVALUATION

In this section, we thoroughly evaluate different sampling strategies in latent spaces for generating music, focusing specifically on (chordified) J.S. Bach Chorales as our main test set. Our main goal is to carefully examine various sampling and interpolation techniques within the latent space, using MusicVAE as our model.

Our evaluation seeks to gauge the effectiveness of various sampling methods in generating chord progressions akin to Bach's original compositions. We compare three sampling strategies as outlined in Section 2: one based on the conventional linear interpolation method, drawn from existing literature, and two innovative strategies centered around

the concept of perceptual relatedness shared by neighboring points in the latent space.

Our hypothesis posits that sampling within a latent space should capture key musical principles such as chord substitutions and parsimonious voice leading. We anticipate a cumulative effect from our strategies: interpolating between adjacent chords introduces variations that harmonize seamlessly with the original musical style. Additionally, employing k-NN enables us to identify chords that share similar traits with the original chord, maintaining harmonic coherence. This approach facilitates the exploration of potential chord substitutions while preserving continuity and smooth voice leading within the progression. This technique is expected to further ensure the preservation of musical continuity and smooth voice leading when coupled with angular displacement to a chord sequence. Moreover, angular displacement is expected to enhance the perceptual similarity to the original chord sequence.

3.1 Materials

To construct a latent space using MusicVAE, we utilize an architecture incorporating a recurrent encoder featuring a two-layer bidirectional LSTM with 1024 units. The decoder is structured hierarchically, employing a two-layer unidirectional LSTM with a hidden size of 1024 for both the conductor and decoder components. A latent size of 256 is selected to significantly reduce input dimensionality while preserving adequate information for effective reconstruction [14]. During the training phase, we utilize the Adam optimizer with an initial learning rate set to 10E-4 and a batch size of 16 to minimize the loss function and refine the model parameters.

As input to our model, we utilize a binary 128-element vector piano roll encoding, representing note activations per unit of time as 'salami slices,' [1] to capture intricate pitch changes within the music.

We employ a standard benchmark of 371 chorales by J.S. Bach to evaluate the proposed strategies, which act as a representative tonal music corpus. The stylistic nature of Bach's chorales makes them an ideal corpus to study harmonic information within the VAE latent space, as the remaining textural aspects are somehow constant, e.g., number of voices, instrumentation, limited change in pitch registers, and harmonic rhythm.

These chorales are obtained from the `music21` library. [2] The corpus comprises chorales in major (53%) and minor (47%) keys, with G Major (14%), A minor (12%), and G minor (11%) being the most prevalent tonalities. On average, each chorale consists of 84 chordified slices. Our training process utilizes 60% of the chorales from the corpus, while the remaining 40% are reserved for testing. To enhance diversity, the chorales utilized for training are subject to augmentation through transposition across all twelve keys, ascending (transposing to the upper 6 keys), as well as descending (transposing to the lower 6 keys).

[1] These slices entail segmented information, capturing the addition or subtraction of pitches from the musical surface each time a change occurs. The salami slice segmentation can be computed from MIDI files by the `chordify()` function within the `music21` software package.

[2] `http://web.mit.edu/music21/`, accessed on March 10, 2024.

3.2 Method

Utilizing the trained VAE latent space, we derive regional chord sequences directly from the original dataset, without transposed augmentation. We gauge the proximity of the three sampling strategies delineated in Section 2, operating under the assumption that the closer they align with the original dataset, the higher their efficacy.

In total, we gather 15276 three-chord sequences from major chorales, with the possible dataset comprising solely chords within major chorales. Furthermore, 13324 three-chord sequences are sourced from minor chorales, exclusively considering chords existing within these chorales for the discrete dataset. Lastly, 28600 three-chord sequences are extracted from all chorales, encompassing all chords within the chorales for the discrete dataset.

We utilize chord distances within a perceptually inspired pitch-space to determine the sampling strategy closest to the original [21]. Here, the Euclidean distances between pitch-class sets reflect their perceptual relatedness [22]. Chords are projected as a chroma vector c_n into a weighted pitch-class Discrete Fourier Transform (DFT) space, using Equation 2, where magnitudes and phases correspond to chord qualities and regional sets. N = 12 is the dimension of the chroma vector, and w_q are weights derived from empirical dissonance ratings of dyads used to adjust the contribution of each dimension q of the space [23].

$$T(q) = w_q \sum_{n=0}^{N-1} \bar{c}_n e^{\frac{-j2\pi kn}{N}}, \quad 1 < q < 6 \qquad \text{with} \quad \bar{c}_n = \frac{c_n}{\sum_{n=0}^{N-1} c_n} \tag{2}$$

In this space, we can compute the Euclidean distance between two given $T_1(k)$ and $T_2(k)$ vectors, representing chords, by utilizing Equation 3 [21].

$$d\{T_1(k), T_2(k)\} = \sqrt{\|T_1(k) - T_2(k)\|} = \sqrt{\sum_{k=1}^{M} |T_1(k) - T_1(k)|^2} \tag{3}$$

We analyze descriptive statistics for the 15276 major, 13324 minor, and 28600 combined chords. The sampling strategy with the lowest mean value serves as an indicator of the closest resemblance to the baseline original J.S. Bach chord sequences.

4. RESULTS AND DISCUSSION

The results of our evaluation are shown in Table 1. Regardless of the distance measures employed and the chorale sets evaluated, chords sampled via the k-NN method consistently exhibit the closest proximity to the original. Following closely are the chords sampled using the k-NN with

angular displacement technique. However, despite our expectations, the angular displacement fails to contribute significant information towards capturing chords that exhibit greater perceptual similarity to the original chord. Linear interpolation consistently yields the poorest results, often exhibiting distance values nearly twice as high compared to the other sampling methods.

		Major	Minor	All
Euclidean Distance	Linear Interp.	19.6 ± 6.8	20.5 ± 6.8	20.1 ± 6.8
	k-NN	**9.6** ± 4.3	**10.2** ± 4.5	**9.5** ± 4.3
	k-NN w/ AD	10.0 ± 4.6	10.7 ± 4.8	9.9 ± 4.5
Cosine Distance	Linear Interp.	19.7 ± 7.2	20.6 ± 7.2	20.1 ± 7.2
	k-NN	**9.5** ± 4.2	**10.2** ± 4.4	**9.4** ± 4.2
	k-NN w/ AD	10.0 ± 4.6	10.8 ± 4.8	9.8 ± 4.5

Table 1. Results of the comparative analysis regarding the perceptual relatedness of the chord sampling strategies. Descriptive statistics for 15276 major, 13324 minor, and 28600 combined chords are presented. Three strategies — Linear Interpolation, k-NN, and k-NN with angular displacement — are evaluated. Distances within strategies, calculated using both Euclidean and cosine measures, are depicted. Best results per dataset of chords are highlighted in bold.

To further inspect the quality of these sampling strategies and enlighten the raised hypothesis that a latent space ought to capture key principles from music theory and practice, such as chord substitutions (by replacing a given chord in a sequence and maintaining the same function) and similar voice leading between adjacent chords in a given sequence, we will study in detail two three-chord sequences from BWV 184.5 and BWV 311, chorales in major and minor keys, respectively.

Figure 4 illustrates a three-chord sequence corresponding to the (authentic) cadence of the initial phrase in J.S. Bach's Choral BWV 184.5 in D Major, along with the five closest candidates selected by each sampling strategy, ordered by closeness to the original. Notably, both the k-NN and k-NN with angular displacement strategies demonstrate comparable selections for four of the five candidate chords, indicating similar hypotheses underlying their choices.

Intriguingly, the third and fifth options are interchanged between the two strategies, while the fourth option varies from k-NN to k-NN with angular displacement. Nevertheless, all options remain within the dominant space, maintaining the function of the original chord.

In terms of voice-leading within the original sequence,

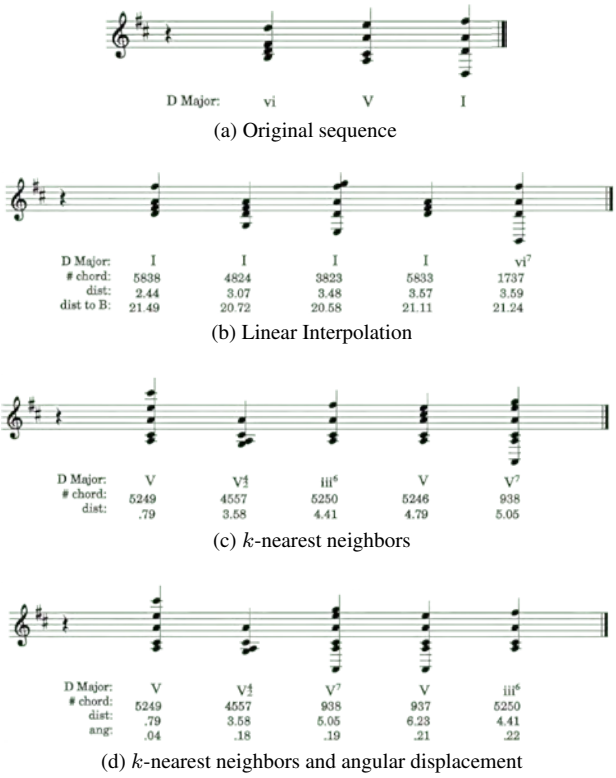
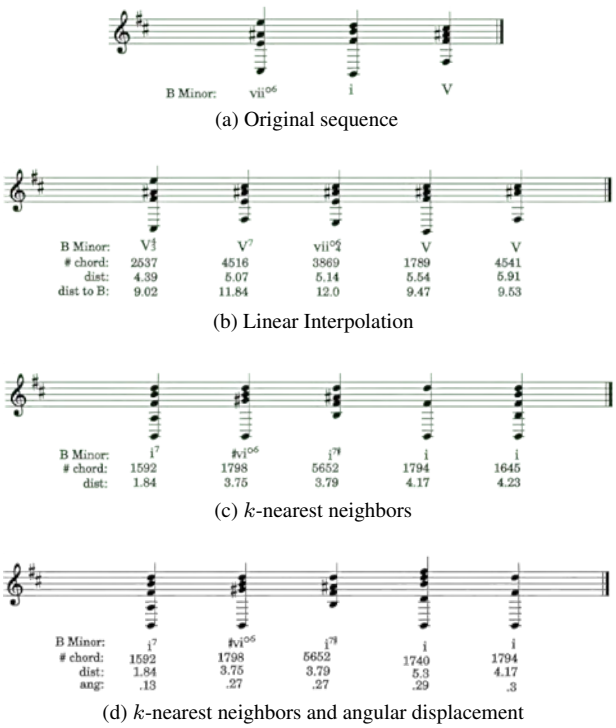


Figure 4. Original sequence, derived from the cadence of the first phrase of J.S. Bach's Choral BWV. 184.5 in D Major, along with 5 optimal sampling candidates using the three different sampling strategies, ordered by closeness to the original. For each strategy, we present the tonal functions within the phrase, the number of the chord in the dataset of all chords, and the distance metric calculated, i.e., in k-NN and k-NN with angular displacement, we present the Euclidean distance (dist) from the candidate point to the original, plus angular distance for k-NN with angular displacement (ang). For linear interpolation, we present the Euclidean distance from the candidate point to the interpolated continue point in the latent space (dist) and the distance to the original chord (dist to B).

contrary to our anticipation, k-NN appears to yield superior bass line interpolation, as the bass notes of the initially sampled chords are notably closer to the first chord. Conversely, diverse candidate chords in k-NN with angular displacement could potentially enhance voice-leading in the upper voices. Even if the bass line leaps from the first chord in the progression to one of these chords, it remains stylistically acceptable, given that Bach's chorales often feature bass line jumps. Additionally, the substantial difference in bass between the first and third chords in the progression suggests that the bass leap from the second chord could effectively prepare the transition to the third chord, aligning with typical movement in Bach's chorales. For instance, both the third and fourth potential selections of k-NN with angular displacement present intriguing options for voice-leading, with the lower and upper voices resolving in opposing directions towards the final chord of the sequence.

In the case of linear interpolation, a contrasting pattern

emerges: all candidates fall within the tonal space of the tonic, notably closer to both the first and third chords than to the second chord in the original sequence. We posit that this phenomenon occurs because the original sequence does not adhere to a linear function, where the middle chord is functionally more distant from both the preceding and succeeding chords. Thus, the interpolated point will tend to remain closer to the starting and ending points of the linear function. Consequently, the movement between voices in all chords is much closer than in the last two strategies, as the notes are mainly the same between the three chords.



(a) Original sequence

(b) Linear Interpolation

(c) k-nearest neighbors

(d) k-nearest neighbors and angular displacement

Figure 5. Original sequence, derived from the cadence of the first phrase of J.S. Bach's Choral BWV 311 in B Minor, along with 5 optimal sampling candidates using the three different sampling strategies, ordered by closeness to the original. For each strategy, we present the tonal functions within the phrase, the number of the chord in the dataset of all chords, and the distance metric calculated, i.e., in k-NN and k-NN with angular displacement, we present the Euclidean distance (dist) from the candidate point to the original, plus angular distance for k-NN with angular displacement (ang). For linear interpolation, we present both the Euclidean distance from the candidate point to the interpolated continue point in the latent space (dist) and the distance to the original chord (dist to B).

In the minor domain, we introduce a second example, displayed in Figure 5. This figure showcases a chord progression referencing a half cadence, ending the initial phrase of J.S. Bach's Choral BWV. 311 in B Minor.

In this scenario, a familiar pattern emerges: the linear interpolation strategy yields candidate chords with voice leading closer to the first and last chords of the original sequence, yet the preservation of tonal function is compromised (in this instance, remaining within the dominant domain). Nonetheless, the difference between these candidates and those in the previous example is less pronounced, with the Euclidean distance to the original chord being notably closer, albeit still farther from the alternatives presented by the other two strategies.

The k-NN and k-NN with angular displacement strategies exhibit comparable outcomes to those in the previous example, with similar options across the two methods, barring one exception. However, between the two different chords proposed by the strategies, the one selected by k-nearest neighbors would likely provide superior voice leading in this specific sequence. This preference arises as the voice leading of the chord suggested by angular displacement would introduce a fourth interval in the upper voice between the second and third chords of the progression, a departure from the natural style observed in Bach's chorales.

5. CONCLUSIONS AND FUTURE WORK

This paper delves into the exploration of sampling strategies within latent spaces for music generation, specifically focusing on chordified J.S. Bach Chorales as a benchmark test set and utilizing MusicVAE as the generative model. We experimentally compare three sampling and interpolation strategies within the latent space to generate chord progressions mirroring Bach's original chord sequences. These strategies depart from an original Bach chorale phrase and evaluate chord substitutions given the previous and sequential chord contexts. Specifically, we employ the following sampling strategies: (1) traditional linear interpolation between adjacent chords, (2) k-nearest neighbors, and (3) k-nearest neighbors combined with angular alignment.

Our approach investigates these sampling strategies' alignment to music theory principles of functional harmony embedding and voice-leading to assess the similarity to the original Bach style and the semantic description of the space. The study yields valuable insights into the dynamics of latent space manipulation for music generation, paving the way for potential advancements in creative autonomy and stylistic fidelity in algorithmic music composition.

Our preliminary findings suggest that k-NN and k-NN with angular alignment to $\vec{AB}$ most closely align with the tonal function of the original chord, especially the former. Interestingly, k-NN tends to offer superior bass line interpolation, while k-NN with angular displacement potentially enhances voice-leading in upper voices. Despite slight variations between strategies, they generally maintain the essence of the original chord progression while offering alternatives with varying degrees of voice-leading quality and tonal function preservation.

Conversely, linear interpolation seems to excel in maintaining aspects of voice-leading. This could stem from its incorporation of both preceding and succeeding chords as reference points, resulting in smoother transitions and minimized leaps in the voices. However, linear interpolation often confines its selections within the tonal space defined

by these two chords. We hypothesize that this tendency may arise from the non-linear characteristics inherent in the original sequence.

These findings underscore the complexity of generating plausible chord progressions within the context of Bach's chorales and highlight the nuanced differences between sampling strategies regarding voice-leading and tonal function preservation. The study suggests that while each strategy offers viable alternatives, considerations such as bass line interpolation, upper voice resolution, and adherence to Bach's stylistic conventions play crucial roles in determining the most suitable chord progression.

In forthcoming research, motivated by the revelation that linear interpolation does not fully capture the original chord progressions in Bach's chorales, we plan to explore alternative interpolation strategies rooted in the broader framework of parametric curves. Our objective is to examine whether the tonal functionality traits exhibit consistent patterns or relationships within these curves. This exploration seeks to uncover more effective methods for modeling chord progressions in Bach's music, thereby advancing our understanding of the underlying structural principles and enhancing the authenticity of generated musical compositions.

Furthermore, we plan to explore deep-learning strategies to identify optimal interpolations (e.g., the Adversarially Constrained Autoencoder Interpolation and the Bridge Process [24, 25]). These methodologies, which have not yet been integrated into symbolic music VAEs, possess the potential to produce more authentic and stylistic interpolations. Their ability to discern interpolation patterns within the original chord progressions suggests an exciting avenue for future exploration. By incorporating these methodologies into symbolic music VAEs, we aim to enhance the fidelity and richness of generated musical compositions, ultimately advancing the state-of-the-art in computational music generation.

In summary, we plan to investigate the applicability of these sampling strategies to other latent spaces within the realm of music, such as audio latent spaces. By extending our exploration beyond symbolic representations, we aim to enhance sampling techniques and improve music generation across various types of models. This broader inquiry holds promise for advancing the field of computational music generation and facilitating the creation of more diverse and expressive musical compositions across different modalities.

Acknowledgments

This research has been funded by the Portuguese National Funding Agency for Science, Research and Technology [2021.05132.BD].

6. REFERENCES

[1] N. Carvalho and G. Bernardes, "Exploring latent spaces of tonal music using variational autoencoders," in *The International Conference on AI and Musical Creativity (AIMC)*, 2023.

[2] ——, "Fourier (common-tone) phase spaces are in tune with variational autoencoders' latent space," in *Mathematics and Computation in Music: 9th International Conference, MCM 2024, Coimbra, PT, June 18–21, Proceedings*, Springer-Verlag, Berlin, Heidelberg, 2024.

[3] N. Bryan-Kinns, "Reflections on explainable ai for the arts (xaixarts)," *Interactions*, vol. 31, no. 1, p. 43–47, jan 2024. [Online]. Available: https://doi.org/10.1145/3636457

[4] K. Chen, G. Xia, and S. Dubnov, "Continuous melody generation via disentangled short-term representations and structural conditions," in *2020 IEEE 14th International Conference on Semantic Computing (ICSC)*, 2020, pp. 128–135.

[5] A. Pati and A. Lerch, "Is disentanglement enough? on latent representations for controllable music generation," *Computing Research Repository (CoRR)*, vol. abs/2108.01450, 2021. [Online]. Available: https://arxiv.org/abs/2108.01450

[6] B. Banar, N. Bryan-Kinns, and S. Colton, "A tool for generating controllable variations of musical themes using variational autoencoders with latent space regularisation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 13, pp. 16 401–16 403, Sep. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/27059

[7] M. Turker, A. Dirik, and P. Yanardag, "Midispace: Finding linear directions in latent space for music generation," in *Proceedings of the 14th Conference on Creativity and Cognition*, ser. CC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 420–427.

[8] Z. Guo, J. Kang, and D. Herremans, "A domain-knowledge-inspired music embedding space and a novel attention mechanism for symbolic music modeling," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 4, pp. 5070–5077, Jun. 2023. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/25635

[9] N. Bryan-Kinns, B. Zhang, S. Zhao, and B. Banar, "Exploring variational auto-encoder architectures, configurations, and datasets for generative music explainable ai," *Machine Intelligence Research*, vol. 21, no. 1, p. 29–45, Jan. 2024. [Online]. Available: http://dx.doi.org/10.1007/s11633-023-1457-1

[10] R. Yang, T. Chen, Y. Zhang, and G. Xia, "Inspecting and interacting with meaningful music representations

using VAE," *Computing Research Repository (CoRR)*, vol. abs/1904.08842, 2019. [Online]. Available: http://arxiv.org/abs/1904.08842

[11] R. Yang, D. Wang, Z. Wang, T. Chen, J. Jiang, and G. G. Xia, "Deep music analogy via latent representation disentanglement," in *20th International Society for Music Information Retrieval (ISMIR)*, 2019. [Online]. Available: https://archives.ismir.net/ismir2019/paper/000072.pdf

[12] A. Pati and A. Lerch, "Attribute-based regularization of latent spaces for variational auto-encoders," *Neural Computing and Applications*, vol. 33, no. 9, p. 4429–4444, Aug 2020.

[13] A. Pati, A. Lerch, and G. Hadjeres, "Learning to traverse latent spaces for musical score inpainting," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, A. Flexer, G. Peeters, J. Urbano, and A. Volk, Eds., 2019, pp. 343–351. [Online]. Available: http://archives.ismir.net/ismir2019/paper/000040.pdf

[14] M. Prang and P. Esling, "Signal-domain representation of symbolic music for learning embedding spaces," *Computing Research Repository (CoRR)*, vol. abs/2109.03454, 2021. [Online]. Available: https://arxiv.org/abs/2109.03454

[15] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *International Conference on Machine Learning*, vol. abs/1803.05428, 2018. [Online]. Available: http://arxiv.org/abs/1803.05428

[16] S. Ji, X. Yang, and J. Luo, "A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges," *ACM Comput. Surv.*, vol. 56, no. 1, aug 2023. [Online]. Available: https://doi.org/10.1145/3597493

[17] L. Mi, T. He, C. F. Park, H. Wang, Y. Wang, and N. Shavit, "Revisiting latent-space interpolation via a quantitative evaluation framework," *Computing Research Repository (CoRR)*, vol. abs/2110.06421, 2021. [Online]. Available: https://arxiv.org/abs/2110.06421

[18] B. V. Dasarathy, *Nearest neighbour norms*. Los Alamitos, CA: IEEE Computer Society Press, Dec. 1991.

[19] D. Huron, "Tone and voice: A derivation of the rules of voice-leading from perceptual principles," *Music Perception*, vol. 19, pp. 1–64, 09 2001.

[20] G. Dong, Y. Qingyun, Shang, Ye, and Y. Zhang, "Direction-aware continuous moving k-nearest-neighbor query in road networks," *ISPRS International Journal of Geo-Information*, vol. 8, p. 379, 08 2019.

[21] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, Oct. 2016, https://doi.org/10.1080/09298215.2016.1182192.

[22] G. Bernardes, D. Cocharro, C. Guedes, and M. E. Davies, "Harmony generation driven by a perceptually motivated tonal interval space," *Computers in Entertainment (CIE)*, vol. 14, no. 2, pp. 1–21, 2016.

[23] F. C. F. Almeida, G. Bernardes, and C. Weiß, "Mid-level harmonic audio features for musical style classification," in *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Bengaluru, India, 2022.

[24] D. Berthelot, C. Raffel, A. Roy, and I. J. Goodfellow, "Understanding and improving interpolation in autoencoders via an adversarial regularizer," *Computing Research Repository (CoRR)*, vol. abs/1807.07543, 2018. [Online]. Available: http://arxiv.org/abs/1807.07543

[25] C. Ringqvist, J. Butepage, H. Kjellström, and H. Hult, "Interpolation in auto encoders with bridge processes," in *2020 25th International Conference on Pattern Recognition (ICPR)*, 2021, pp. 5973–5980.

GUITAR CHORD DIAGRAM SUGGESTION FOR WESTERN POPULAR MUSIC

Alexandre D'HOOGE (alexandre.dhooge@algomus.fr) (0000-0003-1634-3406)[1], **Louis BIGO** (0000-0002-9865-2861)[2], **Ken DÉGUERNEL** (0000-0001-7919-3463)[1], and **Nicolas MARTIN**[3]

[1]**Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL**, F-59000 Lille, France
[2]**Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800**, F-33400 Talence, France
[3]**Arobas Music**, Lille, France

ABSTRACT

Chord diagrams are used by guitar players to show where and how to play a chord on the fretboard. They are useful to beginners learning chords or for sharing the hand positions required to play a song. However, the diagrams presented on guitar learning tools are usually selected from an existing database of common positions and rarely represent the actual positions used by performers. In this paper, we propose a tool which suggests a chord diagram given a chord label, taking into account the diagram of the previous chord. Based on statistical analysis of the DadaGP and mySongBook datasets, we show that some guitar chord diagrams are over-represented in Western popular music and that some chords can be played in up to 108 different ways. We argue that taking the previous position into account as context can improve the variety and the quality of chord diagram suggestion, and compare this approach with a model taking only the current chord label into account. We show that adding previous context improves the F1-score on this task by up to 32% and reduces the propensity of the model to suggest standard open chords. We also define the notion of texture in the context of chord diagrams and show through a variety of metrics that our model improves texture consistency with the previous diagram.

1. INTRODUCTION

On the guitar, a chord can be played in multiple ways, each position having its own pitch, timbral and biomechanical specificities. *Chord diagrams* can be used to represent the position at which a chord is played. They can be notated with a graphical representation as shown in Figure 1, or in text form (compatible with ASCII tablatures) with a number indicating which fret is played on each string. For instance, the chord diagrams from Figure 1 can be annotated, respectively, as `x.0.2.2.1.0`, and `5.7.7.5.5.5`. The former contains both an open (`0`) and a muted (`x`) string and is the most common shape, used by beginners for its simplicity and in pop music for the resonance of open strings. The latter is a *barré* chord and is harder to play, but can be shifted along the fretboard to play other minor chords without changing hand shapes.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

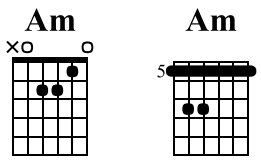


Figure 1: Two guitar chord diagrams for an A minor chord.

We propose the task of chord diagram suggestion to assist composition. Chord diagram suggestion research has so far been focusing on playability [1–3] with limited ability to take into account the relationship between consecutive chords. We can see similar problems in the suggestions of the most popular guitar learning services, which only provide catalogs of standard chord diagrams. While such an approach provides agency to the user, it might also drive beginners towards using the same diagrams over and over again. In this paper, we propose a model that suggests a chord diagram, given a chord label and the previous notated diagram. The contributions of this work are as follows: (i) a context-aware approach for guitar chord diagram suggestion; (ii) a set of metrics to assess performance in this task and characterize texture for guitar chord diagrams; (iii) an illustrative application of this task for rhythm guitar tablature continuation; (iv) openly released code and data for all of the above. The rest of this paper is organized as follows: In section 2, we present related work on symbolic music for guitar in Western popular music repertoire. We then present our proposed approach and models in section 3. In section 4, we discuss in details the datasets and the data augmentation procedure we used. After discussing evaluation metrics, we finally share our results in section 5, before showing how it could be used for rhythm guitar continuation in section 6.

2. RELATED WORKS

A large part of research on guitar focuses on tablatures as a notation system, especially when studying Western popular music. From the audio realm, tablatures are often the output medium for guitar transcription [4–6], but they can even be used as a source of additional data to improve automatic transcription [7, 8]. In a field related to audio transcription, chord recognition from audio is an important task in Music Information Retrieval [9], as attested by its recurring occurrence in MIREX [1] and ongoing research in

[1] https://www.music-ir.org/mirex/wiki/2021:Audio_Chord_Estimation

the neighboring field of multipitch estimation [10]. Chord recognition from audio was also specifically studied in the case of guitar, exploiting knowledge of the instrument to improve algorithms dedicated to it [4, 5, 11]. Once or when available, tablatures can also be both the input and the output, like for symbolic music generation (symbolic referring here to the use of a notation format, like sheet music or tablatures). This topic was already studied using Markov chains [12] and benefited from the advance of deep neural networks [13]. The release of the dataset DadaGP in 2021 [14] also had a significant impact in the field, fostering all kinds of research on guitar tablature generation [15–19]. To assist composition, research has also been conducted on how to *jazzify* chords [20] or generate style-conditioned chord sequences [21]. Tablatures and guitar chord research is also intertwined with the field of guitar fingering research, as deriving tablatures from audio – or scores – requires choosing string/fret combinations to play the notes, which will ultimately be influenced by the fingers used to press the strings and the accompanying biomechanical constraints. This problem has been tackled by a graph-search approach [22] or HMM-like models [23–25] and can then be used for tasks like automatic arrangement [26]. When it comes to guitar chord diagrams however, they are mostly studied with playability considerations in mind [1–3] or in a musicological analysis of their use [27]. In this paper, we consider chord diagram suggestion as an assisted composition task, to help beginner guitar players choose how to play a new chord in a pre-existing sequence, or assist composers in creating guitar accompaniment parts.

3. METHODOLOGY

3.1 Proposed Model

In this work, we suggest a diagram d_t for a chord, based on its label ℓ_t and the previous diagram d_{t-1} (Figure 2).

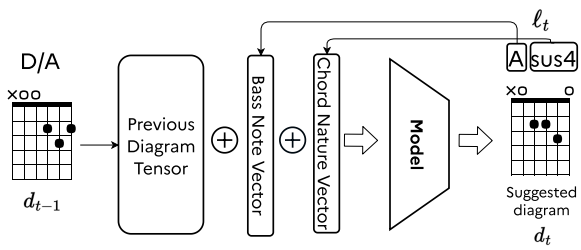


Figure 2: Summary of the proposed approach.

The task consists in finding the diagram which probability is highest in the provided context. To do so, we convert the chord labels and diagrams into vectors and train a neural network on chord pairs. For each chord label, we first extract its bass note, which might be different from its root note in case of inverted chords. This bass note information is then converted into a *one-hot* vector of size 12 where enharmonic equivalents of the twelve-tone equal temperament are merged together (the *Bass Note Vector*). Then, the pitch-class content of the chord is converted into a *many-hot* vector of size 12 (the *Chord Nature Vector*).

Bass Note Vector	$[0,0,0,0,0,0,0,0,0,\mathbf{1},0,0]$	Asus4
Chord Nature Vector	$[0,0,\mathbf{1},0,\mathbf{1},0,0,0,0,\mathbf{1},0,0]$	
	C **D** **E** F G **A** B	

Figure 3: Computed vectors for an *Asus4* chord, played with the diagram shown Figure 2.

An example of such vectors for the *Asus4* chord from Figure 2 is shown Figure 3.

The diagrams are converted into many-hot arrays of size $[6, 25+1]$, each row accounting for a string (6 on a standard guitar) and its 25 frets (counting the open string as a zeroth fret). An additional per-string coefficient is added to account for muted strings. This ensures that all vectors always have a non-zero value for a string, which ultimately permits normalizing predictions and using them as probabilities. In summary, input data contains 180 values for d_{t-1} – the previous diagram – and ℓ_t – the new chord label, from which the bass note and chord nature vectors are computed. As a result, the model outputs 156 probabilities, 26 per string, for d_t.

To evaluate to which extent adding the context information improves chord diagram suggestion, we define the baseline as the same model, but removing information about the previous diagram. In that case, the baseline model takes only 24 values as input, but still returns 156.

3.2 Implementation Details

We use a fully connected neural network as an architecture. Hyperparameters were tuned – through manual exploration – to maximize performance of both the baseline and the proposed model, ending up with no hidden layer for the baseline and one hidden layer of size 150 for the full model. Performance is evaluated with a Binary Cross-Entropy loss on the output, the model using a Sigmoid activation function, and string-wise normalization of the predictions (to enforce a single prediction per string). Training uses the Adam optimizer with $(\beta_1, \beta_2) = (0.9, 0.999)$, and a learning rate $\lambda = 0.001$. The training is stopped whenever the validation loss does not improve by at least $\delta = 0.001$ for two consecutive epochs. Full training of the model can be done in a few minutes on a standard laptop CPU. The Python implementation is openly available, along with a demonstration website, at `algomus.fr/code`.

4. DATA

4.1 Corpora

This study is conducted on two corpora of guitar tablatures containing chord diagram information. The first one is DadaGP [14], a community-based corpus of more than 25000 songs spanning a wide variety of styles, though mostly focusing on rock and metal. The second one is the proprietary dataset *mySongBook* [2] (MSB), which consists of about 2500 songs professionally transcribed.

[2] `mysongbook.com`

The `.gp` file format of these datasets allows including chord diagrams with the tablature data. We only use tracks which contain chord diagram data, reducing the datasets to 2766 and 520 tracks for DadaGP and MSB respectively. From these tracks, we extract chord pairs that occur within a 2 bars interval and finally keep one occurrence of each unique transition $(\ell_{t-1}, \mathrm{d}_{t-1}) \rightarrow (\ell_t, \mathrm{d}_t)$ per track. By doing so, we acknowledge the fact that a chord transition can be more common in a given repertoire and occur in many songs, but we remove duplicate transitions that are inherent to the repetitive nature of Western popular music. This processing step leaves us with 31321 and 7365 transitions for DadaGP and MSB, respectively.

4.2 Statistical Analyses

In this subsection, we provide detailed statistics of the chords used in the datasets. This analysis aims at emphasizing the bias towards more common diagrams and key signatures, due to guitar affordance. Furthermore, because of the considered repertoire, and the large amount of pop/rock/metal tracks, some chord natures are more common than others. However, we also want to underline that this bias is not at the expense of variety, but results in highly unbalanced datasets.

A first dimension to analyze on chords is on what root note they are built. Observation shows that the distributions of root notes are similar between datasets (Figure 4), with more than half the chords having *A, G, D, C* or *E* as root.

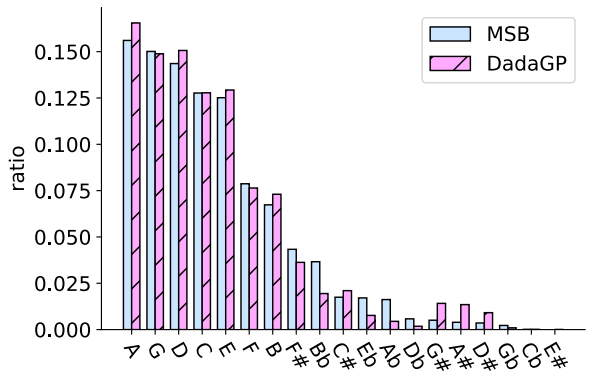


Figure 4: Distribution of the root notes of chords in both datasets.

In both datasets, major chords (M) are the most common (Figure 5), followed by minor chords (m) and *power chords* (5). The tail of the distribution then includes more complex chords, like seventh, suspended or added-tone chords. The distribution of chord natures is in fact highly unbalanced towards the three first classes, making the suggestion of diagrams for less common chords a bigger challenge. Besides, this plot shows that notation between files can vary, MSB containing lots of *7M* and *maj7* labels that are both representing major seventh chords.

Wordcloud representations of the chord labels are also provided (Figure 6) to give an overview of the combined root notes and chord natures. This representation further confirms the unbalanced nature of the datasets but also

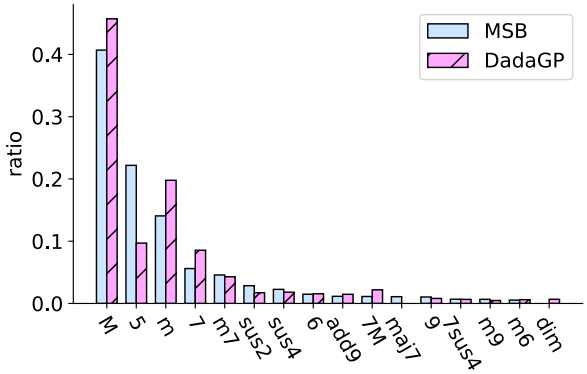


Figure 5: Distribution of the 15 most used chord natures in each dataset.

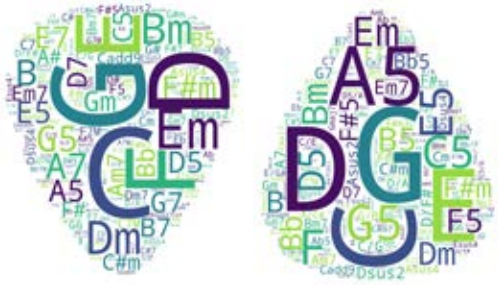


Figure 6: Wordcloud representation of chord labels in DadaGP (left) and MSB (right)

illustrates their differences in the relative importance of chords. For instance, *power chords* are more prominent in MSB than in DadaGP.

While chord labels are already numerous and varied, even more variety comes with the different diagrams that can be used for each label. One can get a sense of it through the median number of diagrams per label: 11.5 for MSB and 24 for DadaGP. This observation illustrates the number of possibilities when suggesting a diagram for a chord, while also showing how a bigger dataset comes with more variety (which could be considered noise). This variety and noisiness of data is particularly noticeable on the most common chords, G Major having 33 and 108 diagrams in MSB and DadaGP, respectively. The variety of diagrams observed comes mostly from the fact that transcribers might write some strings as muted instead of open or played, which can yield different diagrams for the same hand position on the fretboard. The staggering amount of diagrams in DadaGP comes both from the same issue, worsened by the size of the corpus, and occasional incorrect labeling. For instance, some diagrams notated as a G Major can actually represent a *G/B* or a *G5*.

All diagrams observed for each chord label are available, along with the extracted chord pairs, on `algomus.fr/data`.

4.3 Data Augmentation Strategy

The previous statistical analyses showed that a wide variety of diagrams and labels are represented in both datasets. However, more common chords are over-represented and might prevent the model to suggest less used diagrams. We

want to improve the robustness of the proposed approach and reduce its bias towards key signatures more frequent in the datasets so that the model can suggest diagrams even for rare tonalities and chord nature. To do so, we apply the augmentation technique from [12]. For each chord pair, both diagrams are shifted one fret down and the chord labels transposed one semitone down accordingly, until one of the diagrams contains an open string. Similarly, chord pairs are also shifted one fret up until reaching the 15th fret. This maximum is chosen based on the highest diagrams observed in data and to prevent the model from suggesting chords on higher frets, which are used far less in rhythm guitar. Using this augmentation strategy makes the training sets more than three times larger.

5. EXPERIMENTS

5.1 Metrics

Like most tasks on probability estimation, we can evaluate the performance of the model through its Precision (P), Recall (R) and F1-score. However, in the case of this musical task, it could be argued that the so-called ground-truth is not perfect nor the only acceptable answer. For instance, a G major chord can be played with the following diagrams: `3.2.0.0.0.3` and `3.2.0.0.3.3` either repeating the B or the D, and they might be considered equivalent by guitar players. For this reason, we propose hereafter several automated metrics to account for the different errors that can be encountered on diagram prediction.

Pitch Metrics Similarly to [6], we want to measure to which extent the model suggests diagrams that contain all the expected notes for a given chord label. For this purpose, we compute for a chord diagram d the set of pitch classes it contains and compare it with the pitch classes associated with the chord label ℓ.
From these sets, we can compute Pitch Precision ($\mathrm{P_P}$), Pitch Recall ($\mathrm{R_P}$) and a Pitch F1-score ($\mathrm{F1_P}$).

Tablature Metrics Drawing insights from [6], we also define metrics to measure how similar the model suggestions are to the reference diagrams, when comparing them on the fretboard. Similarly to the pitch metrics, we compute the set of string/fret (SF) pairs on the predicted diagram and compare them with those of the expected diagram. We can use those sets to compute String/Fret Precision ($\mathrm{P_{SF}}$), String/Fret Recall ($\mathrm{R_{SF}}$) and the corresponding F1-score ($\mathrm{F1_{SF}}$).

Detect Unplayable Diagrams To evaluate how often the model can return an unplayable diagram, we implement a playability metric inspired by [2]. In this paper, the authors define an *anatomical score* to evaluate the ease of playing a chord. They use a custom fitness function that penalizes chords containing uncomfortable stretches between fingers based on an optimal fingering computed beforehand. This metric allows us to detect unplayable diagrams when obtaining a low anatomical score, like in chords spanning over 5 frets or more. We settled for a threshold of $t = 0.2$ after manual analysis of anatomical scores and playability of the model's suggestions.

Ease of Transition While it is necessary that the suggested diagrams are playable, playability of chord sequences also depends on the chord transitions involved. To assess the ease of the transitions suggested, we also implement the chord change (CC) metric proposed in [4]. This metric analyzes the transition between the diagrams d_1, d_2 through two movements: *wrist movement* m_w, which is obtained as the absolute difference of the index finger fret in the two chords; and the *fingers movement* m_f, defined as the Manhattan distance between the positions of each finger. The final value is inversely proportional to these movements measures and is highest (1) when the chord change is easy ($m \ll 1$) and close to 0 when the transition is very hard ($m \gg 1$).

5.2 Results

For the experiments, we apply a 60-20-20 train, validation, test split to both datasets and average the results over four different splits (Table 1). For fair comparison with the baseline, chord pairs which have identical ℓ_t and d_t (but different d_{t-1}) are considered duplicates and skipped during testing. The data augmentation strategy presented subsection 4.3 is used for the full model but not for the baseline as it decreased its performance. This observation suggests that the results of the baseline should be taken with care, as prediction quality might decrease significantly on data not represented in the training set. The first observation from Table 1 is that the proposed model surpasses the baseline on standard, pitch and string/fret F1-scores. However, both implementations perform well on pitch metrics, showing that the proposed diagrams contain overall the expected pitch content. There is also a significant improvement over the baseline on string/fret metrics, suggesting that information from the previous diagram helps the model choose the correct fretboard area.

		F1	$\mathrm{F1_P}$	$\mathrm{F1_{SF}}$
MSB	Baseline	0.40±.01	0.87±.01	0.46±.04
	Full Model	0.72±.02	0.90±.01	0.67±.02
DGP	Baseline	0.38±.01	0.88±.00	0.45±.02
	Full Model	0.63±.01	0.88±.01	0.60±.02

Table 1: Results of the baseline and the full model on MSB (top) and DadaGP (bottom). Precision and Recall measures were omitted for clarity.

Unplayability and transition ease of the ground-truth and the models' prediction are shown in Table 2. It can be observed that the proposed model suggests unplayable diagrams 15% of the time, which is moderately more than the baseline, with a clearer gap on DadaGP. As a reference, 3-4% of the ground-truth diagrams are deemed unplayable – most of them because the metric does not recognize the *barré* technique with other fingers than the index. However, the proposed diagrams permit slightly easier transitions than the baseline, which is probably again due to the context information that keeps the suggested diagram in the same fretboard area, thus limiting wrist and fingers movements. Nonetheless, even our model with context information has a 10 percentage points difference with the

		Unplayable diagrams	Ease of Transition
MSB	Baseline	0.12±.02	0.15±.02
	Full Model	0.15±.03	0.21±.03
	Test Set	0.03±.02	0.31±.02
DadaGP	Baseline	0.09±.01	0.13±.01
	Full Model	0.15±.01	0.19±.01
	Test Set	0.04±.01	0.29±.01

Table 2: Playability-related metrics on MSB (top) and DadaGP (bottom).

transition ease measured in the datasets, showing that the transitions are still too complicated.

Overall, one can note that performance is similar on MSB and DadaGP. We still decided to share results on both datasets because we deemed relevant the fact that the proposed approach improves diagram suggestion even when training on noisier data. It also shows that a larger amount of chord pairs with an unbalanced distribution does not significantly increase the bias towards common diagrams.

5.3 Diagram Texture Consistency

We want to evaluate here if the *texture* change observed when using a suggested diagram is similar to the one from the reference. To measure this texture, we implement some of the sound quality measures of [2] and variations of them. More precisely, for each diagram, we extract: the ratio of open strings; the ratio of muted strings; the string centroid; and the ratio of unique notes *i.e.* counting only once notes that are repeated on several octaves. We also compute the difference of these metrics from one chord to the next to assess how consistent they are through a transition. The results are reported in Table 3.

All values go from 0 to 1. A first observation is that the performances of the full models trained and tested on DadaGP and MSB are similar. However, this experiment does show slight differences in the datasets, on the amount of muted notes and the string centroids in particular. The biggest difference is the improvement of the baseline's performance on individual chord metrics for MSB, which could be due to the fact that the dataset is smaller and less varied. From the δ values for both datasets, it appears that the metrics are rather consistent from one chord to the next. We can also observe that the baseline trained on DadaGP tends to play too many notes on each chord (lower ratio of muted notes) probably by repeating them on different octaves (lower ratio of unique notes). Finally, an encouraging result is that the model using context suggests diagrams with a texture similar to the ground-truth (lower δ values). It should however be noted that it also tends to repeat more notes than necessary, while still missing some.

6. DISCUSSION

6.1 Application to Rhythm Guitar Continuation

As introduced in [28], a chord diagram suggestion tool could be used in a larger framework for rhythm guitar continuation. Provided with a chord sequence and a tablature

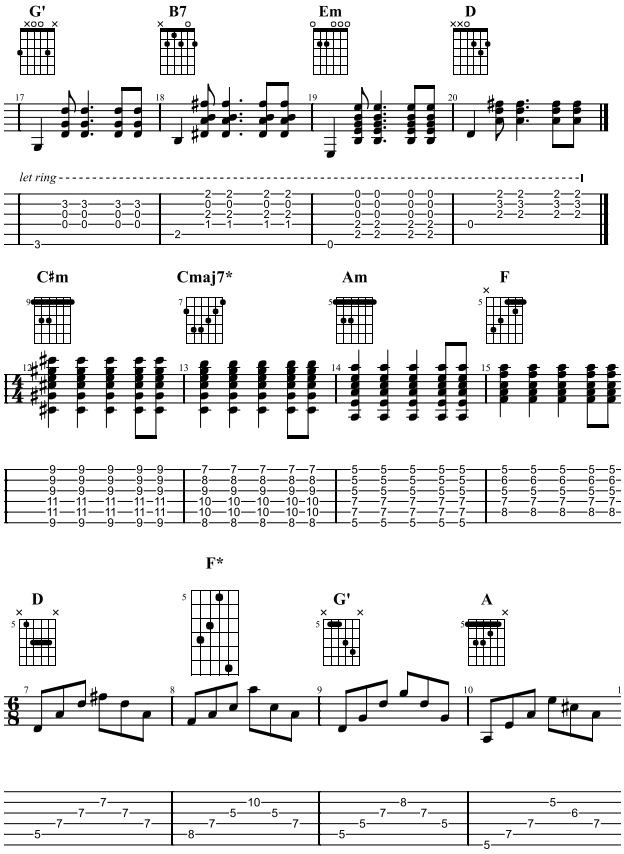


Figure 7: Example of rhythm guitar continuation using the proposed system. The first diagram is chosen randomly, and the next ones are suggested from their label and the previous diagram. Chords with a star are unplayable according to our implemented metric, and chords with an apostrophe do not contain all the notes they should (G5 instead of G).

prompt, a guitar player might be interested in generating the continuation of the prompt. The model introduced in this paper could be used as a first step to choose chord diagrams that are consistent with the prompt, before generating a tablature showing how to strum the new chords. An example of this usage on three different chord sequences is given Figure 7.

Chord label sequences are generated with the tool from [21] and the first diagram is sampled from the probability distribution of all diagrams for the corresponding label. Subsequent diagrams are generated successively using our trained model. To illustrate potential tablature applications, strumming patterns were proposed by the authors for the first bar, and then uniformly generalized to following bars. As discussed in subsection 5.2, the system sometimes suggests unplayable diagrams and does not always include all the notes expected in a chord. However, we think that the sequences are consistent in terms of texture and usually contain diagrams less common than standard open chords and *barré* chords. For instance, in the first sequence, the B7 diagram contains an open string that would help maintain the sound quality of the first chord, unlike the more common *barré* shapes that are `x.2.4.2.4.2` or `7.9.7.8.7.7`. In the second sequence, though the

	DadaGP			MSB		
	BL	Full	Data	BL	Full	Data
Muted notes	0.15 ± .01	0.27 ± .01	0.28 ± .01	0.32 ± .02	0.32 ± .01	0.33 ± .01
$\delta(.)$	0.20 ± .01	0.08 ± .00	0.10 ± .01	0.18 ± .02	0.08 ± .01	0.08 ± .01
Open Strings	0.25 ± .01	0.19 ± .02	0.20 ± .01	0.20 ± .01	0.17 ± .02	0.20 ± .02
$\delta(.)$	0.22 ± .01	0.15 ± .01	0.17 ± .00	0.19 ± .01	0.15 ± .01	0.15 ± .01
String Centroid	0.53 ± .00	0.54 ± .01	0.55 ± .00	0.60 ± .01	0.59 ± .01	0.59 ± .01
$\delta(.)$	0.10 ± .00	0.05 ± .00	0.06 ± .00	0.11 ± .01	0.05 ± .01	0.06 ± .01
Unique Notes	0.61 ± .00	0.71 ± .00	0.80 ± .01	0.68 ± .01	0.72 ± .01	0.80 ± .01
$\delta(.)$	0.21 ± .01	0.13 ± .00	0.12 ± .00	0.20 ± .02	0.13 ± .01	0.11 ± .01

Table 3: Texture metrics for chord suggestions of the proposed model (Full), the baseline (BL) and the corresponding test set (data). $\delta(.)$ denotes the absolute difference of the previous metric between the two chords of a transition.

second diagram is not playable (at least with common guitar techniques), the model suggested keeping a similar *barré* shape and shifting it on the fretboard. However, it changes shape on the last F chord which allows staying around the same frets as the previous Am chord. Finally, the last sequence exhibits a similar behavior, all diagrams starting at the 5th fret, even though the second diagram repeats the A on the 10th fret, making the overall diagram very hard to play.

6.2 Conclusions and Future Work

In this paper, we have shown that chords used in the Western popular guitar repertoire are varied but also highly unbalanced, with some common chords and diagrams being used much more frequently than more complex ones. From this observation, we proposed a new approach to suggest guitar chord diagrams for Western popular music. We showed with several metrics that adding context information through the previous diagram improves the quality of the suggestions while also maintaining a better consistency of texture between chords. All experiments were conducted on two datasets, one proprietary and one public, to further guarantee the validity of the conclusions. Finally, we also gave an example application of the proposed tool in rhythm guitar continuation, where it could help beginners play more interesting and varied chords, or ease the process of writing accompaniment tracks.

Studying this application in more details is a possibility for future work, as well as improving the current suggestion tool, for instance by conditioning the suggestions on musical style. It would also be relevant to increase the number of previous diagrams provided as context. This could allow seeing how far back the system needs to look to maximize the quality of the suggestions, and show when and how continuity in texture is broken. Besides, it is likely that the suggestion system could be improved by musically relevant features on the context like harmony or instrumentation, as other instruments affect greatly the diagrams chosen by guitar players. Finally, while we studied texture consistency through several metrics, it would also be interesting to check how the diagrams suggested affect the voices from one chord to the other and whether the transitions respect some *voice-leading* principles.

Acknowledgments

This work is partially funded under the ANR-TABASCO project: ANR-22-CE38-0001. The authors would like to thank the reviewers, as well as Baptiste Bacot, Louis Couturier, Dinh-Viet-Toan Le, and Mathieu Giraud for their feedback on this paper.

7. REFERENCES

[1] K. Sawayama, N. Emura, M. Miura, and M. Yanagida, "A system yielding the optimal chord-form sequence on the guitar," in *Proc. of the International Conference on Music Perception and Cognition (ICMPC)*, 2006.

[2] K. A. Wortman and N. Smith, "CombinoChord: A Guitar Chord Generator App," in *IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021.

[3] M. A. V. Vásquez, M. Baelemans, J. Driedger, W. Zuidema, and J. A. Burgoyne, "Quantifying the ease of playing song chords on the guitar," in *Proc. of the 24th International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

[4] K. Yazawa, K. Itoyama, and H. G. Okuno, "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[5] E. J. Humphrey and J. P. Bello, "From music audio to chord tablature: Teaching deep convolutional networks toplay guitar," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014.

[6] A. Wiggins and Y. E. Kim, "Guitar Tablature Estimation with a Convolutional Neural Network." in *Proc. of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[7] F. Cwitkowitz, J. Driedger, and Z. Duan, "A Data-Driven Methodology for Considering Feasibility and Pairwise Likelihood in Deep Learning Based Guitar Tablature Transcription Systems," in *Proc. of the Sound and Music Computing Conference (SMC)*, 2022.

[8] Y. Zang, Y. Zhong, F. Cwitkowitz, and Z. Duan, "SynthTab: Leveraging Synthesized Data for Guitar Tablature Transcription," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2024.

[9] M. McVicar, R. Santos-Rodriguez, Y. Ni, and T. D. Bie, "Automatic Chord Estimation from Audio: A Review of the State of the Art," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 2, 2014.

[10] R. M. Bittner, J. J. Bosch, D. Rubinstein, G. Meseguer-Brocal, and S. Ewert, "A Lightweight Instrument-Agnostic Model for Polyphonic Note Transcription and Multipitch Estimation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[11] A. M. Barbancho, A. Klapuri, L. J. Tardon, and I. Barbancho, "Automatic Transcription of Guitar Chords and Fingering From Audio," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, 2012.

[12] M. McVicar, S. Fukayama, and M. Goto, "AutoGuitarTab: Computer-Aided Composition of Rhythm and Lead Guitar Parts in the Tablature Space," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, 2015.

[13] Y.-H. Chen, Y.-S. Huang, W.-Y. Hsiao, and Y.-H. Yang, "Automatic Composition of Guitar Tabs by Transformers and Groove Modeling," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, 2020.

[14] P. Sarmento, A. Kumar, C. J. Carr, Z. Zukowski, M. Barthet, and Y.-H. Yang, "DadaGP: A Dataset of Tokenized GuitarPro Songs for Sequence Models," in *Proc. of the 22nd International Society for Music Information Retrieval Conference*, 2021.

[15] S. Adkins, P. Sarmento, and M. Barthet, "LooperGP: A Loopable Sequence Model for Live Coding Performance Using GuitarPro Tablature," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMUSART)*, 2023.

[16] P. Sarmento, A. Kumar, Y.-H. Chen, CJ. Carr, Z. Zukowski, and M. Barthet, "GTR-CTRL: Instrument and Genre Conditioning for Guitar-Focused Music Generation with Transformers," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMUSART)*, 2023.

[17] P. Sarmento, A. Kumar, D. Xie, CJ. Carr, Z. Zukowski, and M. Barthet, "ShredGP: Guitarist Style-Conditioned Tablature Generation with Transformers," in *Proc. of the 16th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2023.

[18] J. Loth, P. Sarmento, CJ. Carr, Z. Zukowski, and M. Barthet, "ProgGP: From GuitarPro Tablature Neural Generation To Progressive Metal Production," in *Proc. of the 16th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2023.

[19] W. Cui, P. Sarmento, and M. Barthet, "MoodLoopGP: Generating Emotion-Conditioned Loop Tablature Music with Multi-Granular Features," in *International Conference on Computational Intelligence in Music, Sound, Art and Design*, 2024.

[20] T.-P. Chen, S. Fukayama, M. Goto, and L. Su, "Chord Jazzification: Learning Jazz Interpretations of Chord Symbols," in *Proc. of the 21st International Society for Music Information Retrieval Conference*, 2020.

[21] D. Dalmazzo, K. Déguernel, and B. L. T. Sturm, "The Chordinator: Modeling Music Harmony By Implementing Transformer Networks and Token Strategies," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMUSART)*, 2024.

[22] V. Lombardo and D. P. Radicioni, "Guitar Fingering for Music Performance," in *Proc. of the International Computer Music Conference (ICMC)*, 2005.

[23] G. Hori, H. Kameoka, and S. Sagayama, "Input-Output HMM Applied to Automatic Arrangement for Guitars," *Journal of Information Processing*, vol. 21, no. 2, 2013.

[24] G. Hori and S. Sagayama, "Minimax Viterbi Algorithm for HMM-based Guitar Fingering Decision," in *Proc. of the 17th International Society for Music Information Retrieval Conference*, 2016.

[25] G. Hori, "Three-Level Model for Fingering Decision of String Instruments," in *Proc. of the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2021.

[26] S. Ariga, S. Fukayama, and M. Goto, "Song2Guitar: A Difficulty Aware Arrangement System for Generating Guitar Solo Covers from Polyphonic Audio of Popular Music," in *Proc. of the 18th International Society for Music Information Retrieval Conference*, 2017.

[27] J. Cournut, L. Bigo, M. Giraud, N. Martin, and D. Régnier, "What are the most used guitar positions?" in *8th International Conference on Digital Libraries for Musicology (DLfM)*, 2021.

[28] A. D'Hooge, L. Bigo, and K. Déguernel, "Rhythm Guitar Tablature Continuation from Chord Progression and Tablature Prompt," in *Digital Music Research Network One-day Workshop (DMRN+18)*, 2023.

A Sonification Method for Monitoring Chemical Sensor Data

Yutian Hu[1], **Tony Stockman**[1], **Aleksandar Radu**[2], **Ernesto Saiz**[3], **and Nick Bryan-Kinns**[4]

[1]**Queen Mary University of London**, London, United Kingdom
[2]**University of Lincoln** , London, United Kingdom
[3]**Teesside University**, Middlesbrough, United Kingdom
[4]**University of the Arts London**, London, United Kingdom

ABSTRACT

The calibration of chemical sensors is important for ensuring signal integrity. However, for some sensors this can be time-consuming, and without prompt and adequate feedback users may be unaware of calibration errors until post-session analysis is conducted. To address this challenge, we present a real-time sonification framework designed to facilitate efficient monitoring of chemical sensor performance during the calibration stage. The system displays batches of sensor calibration data, collected synchronously, and provides auditory feedback in a sequential time-based manner. To aid users in identifying noisy behaviors in sensor data and comprehending their values, the framework employs three auditory representations: musical note sequences, speech cues, and click sounds. The iterative design process followed a user-centered approach. We detail the iterations of the design, and subsequently evaluate the final approach through a user listening test, discuss benefits and drawbacks of the design, and incorporate user feedback. Our user listening tests conducted using real world data, demonstrate that our method enables efficient detection of abnormal sensor data behaviors.

1. INTRODUCTION

Calibration can be a time-consuming task during the development of novel chemical sensors.

In situations where visual monitoring is restricted or impractical, sonification, the representation of data using sound, provides a solution to the monitoring problem which can be both immediate and intuitive to the user and aid accessibility [1]. In this work, a data sonification system for monitoring the output of chemical sensors without requiring visual interpretation is designed. By transforming certain properties of sensor data into auditory signals, laboratory users are able to gain insights into the condition of sensors by listening to audio without the need for visual representation, thus providing a solution for identifying noise or irregularities in the collected data in real time, that could indicate problematic sensors. The system could also be potentially used to discover data patterns, which may prompt an in-depth analysis on the data at low levels of abstraction.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The primary objective of this project was to explore and develop a sonification system that empowers researchers (users) to easily identify noise or anomalies in data collected during sensor calibration. The user-centered design approach comprised three stages: (1) initial requirement gathering and analysis; (2) iterative design based on expert and user feedback; (3) subjective usability testing.

In the user case in this paper, researchers from University of Lincoln and Teesside University are currently developing a sensor platform which houses up to 16 sensors for continuous monitoring of 4 chemical species in water/soil (4x4 setup, 4 sensors for each analyte). Sensor performance is evaluated based on statistical analysis method, calculation of the slope of the series of data collected from sensors. Data collected from a correctly functioning sensor is expected to have specific patterns. For example, it might show fluctuation at the beginning of each measurement step and it will stabilise towards the end of the measurement.

However, evaluation can be challenging in the early stages under real-world conditions. Sensor signals may deviate unpredictably because of various factors during a calibration session, such as mechanical damage, biofilm formation on surfaces, and the leaching of membrane components. These calibration errors cannot be detected in real-time using current methods, and instead involve processing the data in the laboratory after data collection activities, resulting in a lengthy manual re-calibration process after a certain number of measurements have been taken. In this case, a monitoring method which can help the researchers efficiently detect anomalies during the calibration stage is needed to improve the process and reduce the time spent doing so.

The remainder of this paper is organised as follows: Section 2 provides an overview of sonification methods for monitoring. Section 3 presents the proposed sonification system design process, including the final design, with an emphasis on the user-centered approach applied. Section 4 analyses the results of the user response to a questionnaire for evaluating the sonification system. Section 5 discusses the the main findings of this work. Section 6 concludes the paper by summarizing the findings and the last section listed potential future work. Program code and audio

/ video examples can be found here [1] .

2. OVERVIEW OF MONITORING THROUGH SOUND

Monitoring through sound has a long history and is widely in use. The usage of the stethoscope, a medical device for health monitoring, can be dated back to 4000 years ago in the Ebers Papyrus. More recent studies include monitoring a forest through environmental sound recording with sound information retrieval [2] and a security-monitoring system based on audio classification [3]. Though these applications are quite different, they share the same purpose of helping with the detection of unusual states in different conditions through sound. While these cases tend to apply various analysis techniques to get useful information from real-time data, data sonification creates meaning with sound representations of data to convey reliable information.

The International Community for Auditory Display (ICAD) was established 32 years ago with the goal of fostering discussions and shaping the evolving field of sonification. In that time, more sonification techniques have emerged and been developed. Parameter mapping is a very popular technique allowing for high flexibility in continuous mapping between data and different acoustic properties (e.g. timbre, pitch height, loudness). A 2013 study [4] of sonification mapping strategies found that 86.7% of the projects surveyed utilised variable-pitch mapping, rendering it the most popular approach. Such a mapping strategy also appears frequently in monitoring cases. One moderately successful design for monitoring through pitch is the pulse oximeter, a tiny machine broadly used for measuring heart rate [5]. But even with this approach, the study noted that there are a small number of people who cannot identify the pitch change, thus indicating the potential risk in pitch mapping strategies.

Cases in monitoring through sonification apply different mapping strategies and combine sonification techniques to form an efficient data communication. Andrea et al. [6, 7] propose an Electrocardiography (ECG) sonification system that aims to provide users with unobtrusive auditory cues for monitoring their own or their patients' heart signals through water sounds and morphing timbre. Barra, et al. [8] work on sonification of web servers using personalised musical events to describe web status peripherally. Athina and Andreas [9] present a framework for stock market real-time monitoring through Really Simple Syndication (RSS) feed. Concurrent earcons [10] and synthesised speech are used to represent data.

3. DESIGN PROCESS

With the aim of enhancing the calibration process for chemical sensors, we undertook a three stage user-centred design approach [11] to design and test the sonification of chemical sensors during the calibration stage.

[1] Repository for Pure Data patch, audio sample links and introduction video link in this project: https://github.com/Sparasii-Yuna/Sonification-platform-for-chemical-sensors

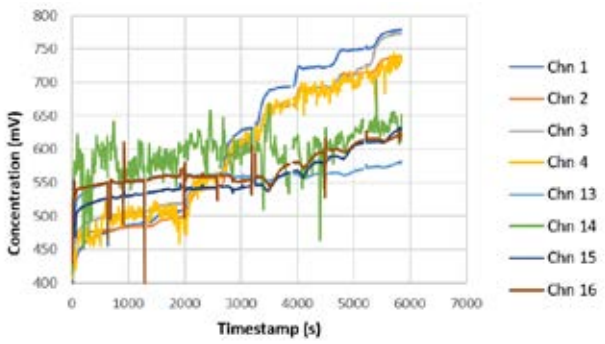


Figure 1. Graph visualisation for example data collected from 8 sensors (Chn1-4 & Chn13-16) during a complete calibration session, combining 9 measurement sessions, with data collected every 8 seconds.

Five participants were involved, including two chemistry researchers with no disabilities (chemistry specialists, also referred to as users in this paper), two experts in sonification, and a sonification designer. Communication between participants included online meetings and emails as well as face to face discussions and focus groups. The design iterations were stored offline in the designer's personal computer and shared throughout the design process. Questionnaires of the final usability listening test are gathered. Audio samples generated in the sonification design were stored on SoundCloud, an online platform for audio content sharing. The three stages of the co-design process are reported in the following sections.

3.1 Stage 1: Requirements Communication

The initial stage consisted of requirements gathering, analysis and early prototyping. The requirements were collected based on comprehensive problem definition documents provided by end users. Additional detail was acquired from verbal explanation on example datasets and research methodologies during two virtual meetings. Example datasets were provided for further analysis.

3.1.1 Current Measurement Precision Estimation Method

Currently there is no real-time analysis by an end user of sensor data - instead, users apply analysis after each calibration session based on statistics and graphs such as in Fig. 1 for sensors performance evaluation.

Details about the statistical analysis method for sensor evaluation are given in [12]. The calculations are derived from the average value of multiple data points collected during each measurement session, where the sensor signal is deemed to be potentially stable. Subsequently, the slope of these averaged values is determined. If this calculated slope falls within the range of what is theoretically considered an appropriate response for the sensor, then the sensor is temporarily classified as reliable. Conversely, if the slope lies outside this range, the sensor's reliability is questioned.

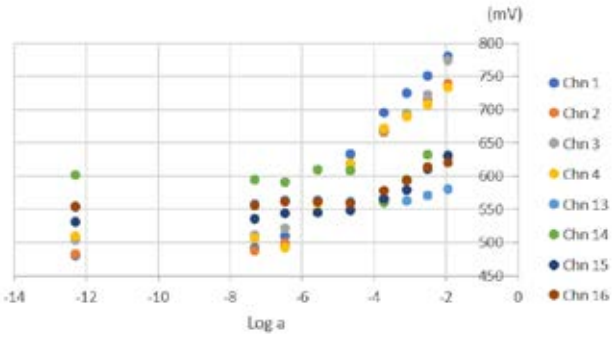


Figure 2. Average concentration values for each of the 8 sensor channels (1-4 and 13-16) across 9 measurement sessions, displayed on a logarithmic time scale.

3.1.2 Dataset

Example sensor data was collected in a laboratory setting. Two different chemical concentrations within a solution were measured with 8 sensors divided into two groups (Group A and Group B), with each group consisting of 4 sensors respectively. On the 4x4 channel sensor platform, Group A was arranged in the first row, occupying slots 1 to 4, while Group B was aligned in the fourth row, taking up slots 13 to 16. For calibration, sensor signals were recorded every 8 seconds over a 10.5-minute period, after which the sensors were removed from the solution. The average of the latest 10 data points which were ideally stablised was calculated for each channel. The most recent data point was excluded due to its tendency to exhibit an abrupt change. This process was repeated across nine measurement sessions, with each session involving an increased concentration level of the target chemical species in the solution. The collected averages from these sessions were then used for the evaluation activities.

Fig. 2 displays all the average values gathered throughout 9 measurement session steps of the dataset. Each slope was calculated through the last 5 points in the graph to ensure that slight deviations at low concentrations are not included. The value of slopes and evaluation results are shown in Table 1. The usability of sensors should be at a good state when the slope value is above 30, theoretically indicating an appropriate sensor response. For example, the data collected from sensor of channel 13 gained a slope value of 4.035, referring to its sensor response being too flat. However, the slope value might fail to reflect the usability. For example, with the slope value of 41.309, sensor channel 14 should be in a good condition. But after observing its signal visualisation in Fig. 1, we could find it extremely noisy, which means the usability is bad and a re-calibration is needed.

3.1.3 Problem Analysis

End users stated that they wish to identify abnormal behaviors in sensor signal as soon as possible during the calibration sessions so that they can take corrective action immediately. Without any real-time monitoring assistance, their current evaluation method relies on calculating the slope

	Chn1	Chn2	Chn3	Chn4	Chn13	Chn14	Chn15	Chn16
Slope	59.104	54.151	55.572	46.400	4.035	41.309	38.273	24.957
Usability (Based on Slope)	good	good	good	good	bad	good	good	bad
Usability (Based on Visualization)	good	good	good	good	bad	bad	good	bad

Table 1. Different usability evaluation results on the same dataset with two methods: slope calculation and visualization observation, showing that the slopes could fail to be used to evaluate the sensors in calibration

of a small number of average numbers from one measurement session, which is highly time consuming. Moreover it is often required to re-calibrate sensors after several measurement steps which compounds the total evaluation effort required. Due to the factors mentioned previously, the sensors are easily affected, thus deterioration in signal is not rare, which makes the situation problematic. Because of this, an efficient monitoring solution is needed to help them evaluate sensor performance with ease and repeatability.

The dataset's structure can be summarized as consisting of 8 parallel abstract one-dimensional series of data. The challenge for sonification lies in analyzing and extracting meaningful variables to represent this data effectively. It is hypothesized that a habitualization process may be necessary for sonification, indicating the need for a systematic approach to familiarize individuals with the auditory representations of the sensor data over time to optimize understanding and interpretation.

Analysis of the problem requirements thus motivates our sonification research in two ways:

- How can we employ sound as an effective means to tell when unacceptable deterioration appears in the sensor signal?
- How can sound inform us of the approximate concentration of the target ion?

3.2 Stage 2: Iterative Design

Iterative design of sonification of sensor data was driven by expert evaluation and user feedback. Design iterations are described in Table 2 and audio samples are available on SoundCloud [2]. During this stage, three online design review meetings were held to gather feedback and ensure the system was meeting the problem requirements. The first meeting evaluated a conceptual design resulting from the previous requirements communication stage and gathering suggestions from experts. The design was developed after the abandoned prototype V0. The design suggested sequential display of data from different channels and mapping data to musical notes instead of direct frequency mapping. A sonification platform was then developed with Pure Data [3] once the specific structure of conceptual design was agreed upon. The second meeting and the third meeting both focused on listening tests and feedback gathering from participants of the meetings. Audio samples were generated of V1 and V2 and shared in advance of

[2] Audio samples of iterations: https://on.soundcloud.com/4xkLX
[3] Pure Data official website: https://puredata.info/

the second meeting for user listening test, while the listening test for V3 took place in the third virtual meeting via a shared screen to share the audio and Pure Data interface simultaneously, so that sound-parameter mappings could be observed in real-time.

Expert evaluation and usability testing with actual users were applied in each prototyping cycle. Experts listened to outputs from our prototype systems and evaluated the quality and usability. For example, during the second meeting, to evaluate the V1 and V2 design, both experts mentioned that the amount of reverb effect in the audio samples of V1 was too much, leading to an optimization for user experience in the next iteration.

3.2.1 Audio Samples

In this section we introduce the audio samples on SoundCloud of different versions of prototypes. The description of representations applied in each iteration is listed in Table 2.

The "V0 FastPrototyping" audio sample, derived from an early prototype during the requirement gathering stage, mapped data values to frequency for simultaneous data presentation. However, due to the challenge of identifying the data source channel in the audio stream, the design was ultimately abandoned.

The audio samples "V1 Chn1-4", "V1 Chn13-16", "V2 Chn1-4", and "V2 Chn13-16" were reviewed in the second meeting. V1 and V2 implemented two variations of design where a fixed set of continuous data was represented as sequences of musical notes following a speech cue, displaying the sequence of four channels individually. It's worth noting that Chn1-4 are for data from well-performing sensors while Chn13-16 are from sensors including problematic ones.

The samples "V3 Chn13-16 Overlapped" and "V3 Chn13-16 NonOverlapping" are examples of the audio generated in the third meeting, where participants observed real-time parameter changes in the system controlled by the designer. the "Overlapped" version had speech cues overlapping with note endings, while the "NonOverlapping" version avoided this overlap.

The final design samples, "V5 AlertRange 10 Chn1-4" and "V5 AlertRange 10 Chn13-16", contrast with "V5 AlertRange 5 Chn1-4" and "V5 AlertRange 5 Chn13-16", which feature a lower alert threshold set at 5. More details will be introduced in the next section.

3.3 Final Sonification Design

The final design employs three types of audio representations as described in this section: musical note sequence (pitch representation), speech cues and click sound.

3.3.1 Musical Note Sequence

The pitch of each musical note is mapped to values of the data, with a higher pitch being related to a larger value, as humans are naturally sensitive to pitch [13], and have the ability to tell whether one tone is higher than another [14]. The motivation is that many listeners should be able to infer a change in value by perceiving difference between two notes, and grasp the data contour through the sequence (melody) contour.

Version	Representations	Description	Feedback
V0	pitch	4 pitch, same timbre, display simultaneously	[abandoned] unable to identify each stream
V1	musical note sequence speech cues	5-note sequence after speech cue, notes with same timbre, short interval between notes, with reverb	too much reverb
V2	musical note sequence speech cues	5-note sequence after speech cue, notes with different timbres (different wavetables), long interval between notes, no reverb	each note can be heard clearly difference between timbres not significant
V3	musical note sequence speech cues	5-note sequence after speech cue, notes with same timbre, short interval between notes, with less reverb, more adjustable intervals	suitable reverb and suitable number of notes need to highlight big changes, ask for no overlaps between speech cue and notes
V4	musical note sequence speech cues click sounds	5-note sequence after speech cue, notes with same timbre, short interval between notes, with less reverb, click sounds with slightly different timbres with granular techniques	[abandoned] click sounds too random
V5 (Final)	musical note sequence speech cues click sounds	5-note sequence after speech cue, notes with same timbre, short interval between notes, with less reverb, click sounds with the same timbre	user feedback (see Section 4)

Table 2. Iterations During Middle Stage Design Process (developed in Pure Data)

We arrange 5 musical notes in a sequence to represent 5 successive data items (from oldest to most recent) collected from one sensor. Each sequence is played with an inter-onset interval (IOI) setting as 150 ms, which could translate to a total sequence duration of 600ms. This timing was chosen to enable listeners to clearly discern the pitch of each individual note, while also ensuring the sequence is brief enough to accommodate the playback of four sequences within an 8-second period — the interval between updates in sensor data. As the dynamic range and value of original data is too large for directly mapping to musical notes, data input is scaled down with an offset to map to notes in a range within the human audition. The scaling strategy also aims to align the pitch variations with dynamic in data. The mission is to keep the pitch at a stable level (with little or no pitch changes) when the data becomes stable, but also to ensure that the musical sound acting unstable (with frequent pitch changes in larger intervals) when the data changes substantially in value. Regarding the timbre of the musical notes, pure tones generated by sine waves with reverb effect are adopted in the final design. Ultimately, a scale factor of 1:25 is applied.

3.3.2 Speech Cues

Four speech cues are used to represent the index of the current sequence about to be played. The speech cues are female speech of four numbers ("one", "two", "three", "four") recorded from the text-to-speech service of Google Translate [4]. With groups of four channels of data to be sonified, speech cues help listeners identify the data source of a sequence. A speech cue is played in advance of the tonal sequence it refers to. The time interval between the onset of a speech cue and its associated sequence is set at 600 ms as this interval is short enough for listeners to group the speech cue with its relevant sequence. It is also long enough to prevent overlap between speech cue and note sequence. However, it was found during the sixth meeting that such overlap can cause annoyance and confusion

[4] https://translate.google.com/

to users. Fig. 3 visualises how the sonification combines speech cues and note sequences in a group of of sensor data channels for five data points per channel.

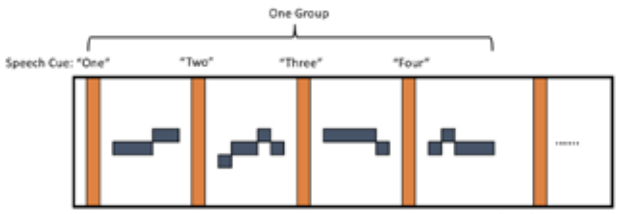


Figure 3. Visual Representation of Speech Cues and Musical Note Sequences

3.3.3 *Click Sound*

Click sounds function as alert audio signals highlighting abnormal behaviors in the data. As an auditory icon, this representation is intended to be easy for users to learn. An abnormal behavior is defined as the difference between successive data exceeding a fixed threshold (alert range). However, the appearance of abnormal behaviors does not-necessarily mean that the sensor is problematic, but a high frequency in abnormal behaviors can imply potential problems. Fig. 4 shows a pair of examples about the mechanism of click sound triggering.

Figure 4. Two click sound mechanism examples. Data is taken from the dataset described in Section 3.1.2.

In the final design the alert threshold is set to 10 based on subjective evaluation of the designer and analysis of 4 example data samples.

Fig. 5 illustrates the frequency of click sounds when the alert range is set to 5 (lefthand of figure) and 10 (right-hand of figure). Eight data samples are illustrated in the figures with time axis ascending vertically downwards. An alert range threshold of 5 shows a high frequency of false-positive alerts (regular clicks) in Chn4 (the fourth column from the left), whereas an alert range threshold of 10 does not include these false-positives, suggesting that a threshold of 10 for the alert range reduces the false-positive alerts.

3.4 Stage 3: Evaluating The Final Design

Usability testing of the interactive sonification was undertaken via listening test with a structured questionnaire and an extra interview. As users may be not familiar with the style of listening required for the sonification design, an introductory video was made to explain the mechanics and

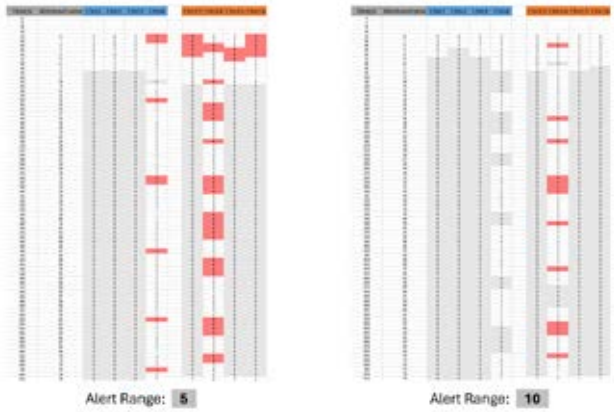


Figure 5. Comparing different alert range threshold settings, 5 (left) and 10 (right). If the number of clicks in a cell is 4 (maximum data points in a sequence), the cell is marked red. If the number is 0, indicating no occurrence in a sequence, the cell is marked as grey.

provide guidance. The video introduced audio representations and their relationship with data. Short audio excerpts that were typical of the system were provided with visualisation and explanations to aid understanding. A questionnaire consisting of 6 questions was administered to the two users to gather qualitative information about usability of the design, particularly in identifying problematic sensors. These questions are listed below:

- *Do you think this design helpful for monitoring data during a calibration session?*
- *Can you observe abnormal behaviours in data streams through this design? If yes, do you think you rely more on pitch information or click sound?*
- *Do you think that the click sound mechanism can accurately reflect the severity of problems in data stream?*
- *Can you easily deduce from this sonification whether there are any problematic sensors within a group and subsequently identify them?*
- *Do you feel it hard to notice any changes in sonified data after listening to it for a period of time?*
- *Any other suggestions/questions/description of feeling?*

4. RESULTS

4.1 Results of Final Stage Evaluation

First we report on the responses to the questionnaires by our two users (U1 and U2) to assess the performance of our sonification system. Both users are familiar with the dataset and the laboratory setting and so provide domain expert feedback on the suitability of the sonification, but they are not experts in sonification and are not musically trained.

Generally the design is regarded as potentially helpful for data monitoring during a calibration session by the users. Both users reported a positive attitude towards the design, evaluating the design as "*in principle helpful*" (U1) and with "*good potential*" (U2).

However, some problems in the design were also identified. One user stated that "*perhaps four sensors are too many*" (U2) which indicates that the information provided in a group exceeded his cognitive load. Beyond that, a correct understanding about the relationship between musical notes and data may have not have been established in users' mind, which may have introduced some misleading interpretations when users were listening to the sample materials. Both users appear to have expected a stable signal to have less variations in pitch, as suggested by the response "*I expect much less variations*" (U2) for stable signal and "*the pitch may be going up or down (due to the jump in value over the threshold) depending on the type of ion*" (U1). These statements may indicate that there is some misunderstanding about the relationship between pitch and data. To be specific, U1 used "*jump in value over the threshold*" to explain the cause to change in pitch / note, which does not match the nature of scaling. With the scaling technique applied, data in specific ranges are mapped to specific notes. For example in our design, with a scaling factor of 1:25, data ranging from 300-325 all map to the musical note G#3. If the data is subtly changing but also changing across the boundary between two specific ranges, the change in pitch would also happen in a sequence. It will act as a "*pitch going up and down*" (U1), which may conflict with users' expectation for a stable pitch in stable signal as their understanding appears to be that change in pitch only happens when jump in value is over certain threshold.

Compared to pitch, click sounds may have improved the performance of the sonification and may be easier for our users to understand. One user "*was able to identify a problematic sensor*" (U2) once they "*were able to focus on it (the click sounds)*" (U2). they "*almost entirely ignored pitch and focused on the click sound*" (U2) and they stated that they are probably "*not able to identify the problematic sensor using the pitch alone*" (U2), which suggests that the click sounds deliver clearer information than pitch to the users. U1 did not express a strong preference for click sounds through their responses, but stated that "*click on a regular basis could mean that the sensor is not working properly (possible failure)*" and "*If only one or two of them are getting the clicks, we may be able to discard its data, or replace them in the field if necessary*". Through these statements, we suggest that U1 understood the representation and was also able to gain more insight into sensor conditions through listening to click sounds.

We found that training is needed in this design to help users familirise themselves with monitoring through sonification. Both users mentioned the importance of training and one user stated that the key is to know "*how it sounds when the sensor is failing and how it is a sensor that it is working*" (U1). We suggest that our introductory video functioned as a useful training guide. Both users relied on the video to make sense of the design. One user watched the video "*many times just to understand the connection between signals in visual terms versus in sound terms*" (U2). The other user commented that the video "*provides the information needed to understand the sonification of the data*" (U1). However, more effort might be needed to explain the relationship between pitch and data clearly. As U2 said, "*I am afraid that I still do not have a full comprehension of how pitch helps*". U2 also suggested "*a slower introduction*" — despite multiple viewings, U2 still felt confused by the musical note sequence mechanism. This implies that the introductory video is helpful but insufficient on its own. To enhance user comprehension, a more detailed and interactive training approach or a longer training session might be beneficial.

4.2 An Extra Interview Session

As some false observations were reported in U1's response, an extra meeting with U1 was held to walk through their responses and discuss the results in more detail. Here the false observation refers the user reporting that they heard click sounds where there were no clicks in the sample audio. The feedback sheet from U1 was also updated after this meeting. During the interview, it was confirmed that the reason for false observations was that user guessed the sonification results with their prior knowledge of the sensors. It was also noted that, besides Channel 14 (Chn14), two additional sensors associated with Channels 13 (Chn13) and 16 (Chn16) were identified as problematic. However comparing to the sensor relating to Chn14 (acting extremely noisy when collecting data), the other two sensors were very stable but did not reflect correct data values as expected. Such failure in sensors cannot be detected by click sounds, as the purpose of click sound representations is to indicate behaviors of large jumps in values of successive data, to inform the users that the signal is noisy. Despite all this, after a discussion with U1, it was shown that this kind of behavior can still be observed through pitch. As U1 stated in the updated feedback: "*After the additional steps during the calibration process two sensors are keeping similar values or going downwards instead of upwards. So, there are no changes in pitch, when there should be.*" Meaning that through pitch representations, users can also potentially identify failing sensors which are not providing responses.

During the meeting, the misunderstanding from U1 about the pitch mechanism was also discussed and corrected. It was found that the users did not manage to acquire enough knowledge about pitch representation through the introduction video, thus leading to an incorrect understanding about the scaling process. U1 suggested adding more details in the instructions to explain the sonification process better, particularly the assignment of each pitch to specific data ranges. After understanding this relationship, U1 realized that pitch information reflects the concentration range of the sample and found the representation more useful. Additionally, U1 concluded that they would rely more on the click sound to identify regular noisy sensors.

5. DISCUSSION

In this section discussions focus on relating the findings in results with the two research questions proposed before.

Our design of click sounds mechanism can effectively inform users of abnormal deterioration in signal. Results of the final usability test shows that both users are able to detect anomaly with click sounds. After a small amount of training, users easily recognised click sounds in an audio stream where musical notes and click sounds are heard simultaneously. What's more, one user reported that only click sounds can be identified after listening for a long time. One possible reason for this might be that the meaning of note sequences was not as clear as click sounds for users. Similar results are also found by Edworthy et al. [15] showing that auditory icon alarms are easier than tonal alarms to be identified either in priority and function domain. As it is easy to adjust parameters (e.g., alert range) in real-time within the Pure Data patch of this work, users could customise the sonification in laboratory settings to tailor it to different needs,

The design of musical note sequence can potentially inform users of the estimated level of target ion's concentration when user has a solid understanding of the mapping strategy. From our results, we suggest that the display of note sequences from 4 sensors in 8 seconds may lead to saturation of cognition load in users quickly, especially with limited training. Compared with the design of the pulse oximeter [5], a successful sonification design which uses pitch to monitoring data from one channel, our sonification displays data from four channels one by one in short delay. The continuity in monitoring one channel is interrupted. Though the trend in data can be observed in the beginning by paying attention to note sequence, users reported to get lost shortly after. It might be because more attention were paid to click sounds during the listening test as the task of identifying noise was more urgent. It was also found that when a tone sequence shares the same contour and a similar tonality with the other sequence just displayed, it will be hard for listeners to discriminate between them [16].

5.1 Future Work

The proposed sonification method in this work may benefit from some additional experiments to fine tune its behavior and improve user satisfaction. Our evaluation was based on the same dataset. Evaluation on different kinds of data inputs would be useful to know and determine if the sonification approach can be generalised to other kinds of data. More experiments need to be carried out to find reliable parameter settings, for example, the setting of the Inter-onset interval (IOI) of note sequences and intervals between groups. More usability tests with a larger focus group with customisable parameters need to be conducted. Finally, experiments investigating concurrent auditory icon and tone sequences would be useful based on this work.

6. CONCLUSION

We suggest that our final sonification design strategy can efficiently help users monitor some abnormal behaviors (noisy activities) in real-time data streaming and identify problematic sensors. Our results support the potential for sonification to aid in these chemical sensors monitoring tasks, and indicate a need to further validate our sonification approach by running them in the real monitoring tasks. During the final usability test, it was found that the click sound representation makes the sonification more useful in identifying the sensors that are acting too noisy, while the pitch representation makes the sonification potentially useful in identifying sensors with inadequate responses. Training sessions may improve the performance of the sonification method by familiarising users with listening modes for monitoring tasks. The introductory video in the final design evaluation was helpful for users to learn how to listen to the sonification but also requires some improvement in explanation of pitch representation. Crucially, however our user-centered design process with participatory design has yielded a successful sonification design with end-user satisfaction. Such a design process can be effective in interdisciplinary scenarios. Above all, our sonification design may potentially be very useful in real-time monitoring task of data collected by chemical sensors.

7. REFERENCES

[1] J. Lazar, S. Chakraborty, D. Carroll, R. Weir, B. Sizemore, and H. Henderson, "Development and evaluation of two prototypes for providing weather map data to blind users through sonification," *J. Usability Studies*, vol. 8, no. 4, p. 93–110, aug 2013.

[2] E. Olteanu, V. Suciu, S. Segarceanu, I. Petre, and A. Scheianu, "Forest monitoring system through sound recognition," in *2018 International Conference on Communications (COMM)*, 2018, pp. 75–80.

[3] A. Abu-El-Quran and R. Goubran, "Security-monitoring using microphone arrays and audio classification," in *2005 IEEE Instrumentation and Measurement Technology Conference Proceedings*, vol. 2, 2005, pp. 1144–1148.

[4] G. Dubus and R. Bresin, "A systematic review of mapping strategies for the sonification of physical quantities," *PloS one*, vol. 8, no. 12, p. e82491, 2013.

[5] G. T. Schulte and F. E. Block, "Can people hear the pitch change on a variable-pitch pulse oximeter?" *Journal of clinical monitoring*, vol. 8, pp. 198–200, 1992.

[6] A. L. A. Blanco, S. Grautoff, and T. Hermann, "Cardiosounds: Real-time auditory assistance for supporting cardiac diagnostic and monitoring," in *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*, ser. AM '17. New York, NY, USA: Association for Computing Machinery, 2017. [Online]. Available: https://doi.org/10.1145/3123514.3123542

[7] A. L. A. Blanco, S. Grautoff, and T. Hermann, "Heart alert: Ecg sonification for supporting the detection and diagnosis of st segment deviations," in *5th Interactive Sonification Workshop (ISon 2016)*, 2016, pp. 1–8.

[8] M. B. T. C. A. De, S. U. F. Petrillo, A. Negro, V. Scarano, T. Matlock, and P. P. Maglio, "Personal webmelody: Customized sonification of web servers," 2001.

[9] A. Bikaki and A. Floros, "An rss-feed auditory aggregator using earcons," in *Proceedings of the 6th Audio Mostly Conference: A Conference on Interaction with Sound*, ser. AM '11. New York, NY, USA: Association for Computing Machinery, 2011, p. 95–100. [Online]. Available: https://doi.org/10.1145/2095667.2095681

[10] S. A. Brewster, P. C. Wright, and A. D. N. Edwards, "Experimentally derived guidelines for the creation of earcons," 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:17462746

[11] T. Robertson and J. Simonsen, "Participatory design," *Routledge international handbook of participatory design*, p. 1, 2012.

[12] P. W. Dillingham, T. Radu, D. Diamond, A. Radu, and C. M. McGraw, "Bayesian methods for ion selective electrodes," *Electroanalysis*, vol. 24, no. 2, pp. 316–324, 2012.

[13] L. Demany and F. Armand, "The perceptual reality of tone chroma in early infancy," *The journal of the Acoustical Society of America*, vol. 76, no. 1, pp. 57–66, 1984.

[14] J. Plantinga and L. J. Trainor, "Memory for melody: Infants use a relative pitch code," *Cognition*, vol. 98, no. 1, pp. 1–11, 2005.

[15] J. R. Edworthy, C. J. Parker, and E. V. Martin, "Discriminating between simultaneous audible alarms is easier with auditory icons," *Applied Ergonomics*, vol. 99, p. 103609, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687021002568

[16] W. J. Dowling, "Tonal strength and melody recognition after long and short delays," *Perception & Psychophysics*, vol. 50, pp. 305–313, 1991.

INTERACTIVE MUSICAL PERIODIC TABLE: SONIFICATION OF VISIBLE ELEMENT EMISSION SPECTRA

Walker SMITH (walksmit@alumni.iu.edu) [1, 2]

[1] *Center for Electronic and Computer Music*, **Indiana University Bloomington**, IN USA
[2] *Department of Chemistry and Biochemistry*, **Indiana University Bloomington**, IN USA

ABSTRACT

Is the periodic table a musical instrument? While this may seem like an absurd question, the process of data sonification can be used to convert the visible spectra of chemical elements into sounds. Each element releases distinct wavelengths of light depending on its electron energy levels—a sort of "chemical footprint" unique to every element. These frequencies of light, which we perceive as different colors, can be represented as frequencies in the audio domain, allowing us to "hear" each distinct color as a sine wave with a unique frequency. This research project involved the construction of an interactive musical periodic table, combining visual representations with sonifications of elemental spectra from high-resolution spectral datasets. Implemented in Max/MSP and Jitter, the program provides audiovisual displays of each element from its rich spectral data. The user can listen to all spectral lines of an element simultaneously (chord mode), or individual lines in succession (melody mode). Future work will involve implementation of this tool in K-12 classrooms to evaluate its efficacy as an interdisciplinary teaching tool bridging STEM and the Arts.

1. INTRODUCTION

The field of chemical spectroscopy concerns the study of atoms and molecules using electromagnetic radiation. Connections between chemical spectra and acoustic spectra have been made by Delatour, who proposed methods for the auditory representation of molecular vibrational spectra [1]. Sonifications of molecular vibrational spectra have been employed by chemical education researchers to develop tools for auditory analysis of chemical spectra for blind and visually impaired students [2]. Nuclear Magentic Resonance (NMR) Spectroscopy is another technique that has enjoyed a productive relationship with music and sound; NMR sonifications have been applied to chemical education [3] and auditory analysis of spectra [4]. NMR and other spectral data has also provided inspiration and musical material for composers [5], including in prior work by the author in "The Sound of Molecules" [6], where sonifications of NMR spectral data are accompanied with narration to create a "sonic tour of the molecular world.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The analytical techniques discussed above use regions of the electromagnetic spectrum that are not visible to humans (infrared and radio-frequency radiation for molecular vibrations and NMR, respectively). However, atoms and molecules can also absorb and emit visible light. When stimulated by a large amount of energy, atoms produce visible emission spectra–vibrant collections of colors resulting from electrons within atoms changing energy levels [7]. Sonifications of these spectra can be accompanied by the corresponding visual spectra as a tool for audiovisual composition. This is demonstrated in the author's composition "Chromatic Chemistry," in which the spectrum of helium is sonified and used to create an audiovisual "Helium Dance Party" [8]. These multimodal representations could also potentially be used for spectral data analysis and/or chemical education.

The Interactive Musical Periodic Table presented here is a natural extension of the author's prior work creating live music/science shows based on chemical sonifications like "The Sound of Molecules" [9]. The goal of this project was to design a prototype of an interface that both the author and others could use to listen to sonifications of the visible spectra for all the elements in the periodic table.

2. METHODS

2.1 Mapping light to sound

Analogies between light and sound extend so far into scientific history that Newton's assignment of the seven discrete colors (red, orange, yellow, green, blue, indigo, and violet) to the continuous light spectrum was done so that the number of colors would match the number of tones in the Western 7-note diatonic scale [10]. Newton's color circle published in his *Opticks* [11] associates the colors with the notes of a D dorian scale, beginning with D (red) and ascending to C (violet) as shown in Figure 1.

Remarkably, even though Newton did not measure the wavelengths of the different colors of visible light (these were not determined empirically until 1802 by Young [12]), the relationships Newton devised are remarkably close to reality; the ratios of wavelengths (and thereby frequencies) of different colors of light and the ratios of the corresponding musical notes to which Newton associated them are more or less correct. This is because the visible spectrum spans very nearly one "octave" of light; the highest frequency light (violet) has roughly double the frequency of the lowest-frequency light (red). Thus, associating the visible spectrum of light with an ascending scale

Figure 1. Figure from Newton's *Opticks* illustrating his association of notes of the D dorian scale with the colors of visible light.

from red to violet is a very close approximation of what colors would sound like if we could listen to electromagnetic radiation.

Of course, even if we pretend that we can "hear" the electromagnetic waves that constitute light, the frequencies of visible light waves range from 380-750 THz, which is on the order of 10^{11} to 10^{13} times larger than the frequencies humans can hear. Multiplying the frequencies of light by a scaling factor in the range of 10^{-11} to 10^{-13} places them comfortably into the human hearing range of 20-20,000 Hz. Scaling the frequencies of visible light like this is equivalent to conceptualizing of them as a musical scale and transposing it into an octave that humans can hear.

As the human hearing range spans 9-10 octaves, this gives us a choice of which octave of sound we choose to place visible light (and therefore, the visible spectra of the elements). Different factors may lead a scientist or composer to choose a certain octave in which to place visible light; for this work, a combination of practical and aesthetic decisions guided the selection of the range of 380-750 Hz (obtained by multiplying the light frequencies by the scaling factor of 10^{-12}) as the default setting for sonification of elemental spectra. This frequency range (roughly corresponding to the middle of the piano, approx. F#4 to F#5), sits within the range of 250-4000 Hz, where the human ear has the greatest frequency discrimination (the average normal hearing adult can discriminate frequency differences of 0.2-0.3%) [13]. This practical consideration is important as two goals of these sonifications are for users to be able to (1) discern differences in elemental spectra purely by listening and (2) associate differences in sound with visual differences in the spectra. This range of 250-4000 Hz still contains several octaves, so an aesthetic decision guided further selection. Namely, at the lower end of this range, the spectra are more pleasant to listen to. When placed in a higher part of this range, such as the octave of 2,000-4,000 Hz, the spectra with hundreds or thousands of distinct lines can result in beating patterns that are more grating and unpleasant than they are in the lower range of 380-750 Hz. Thus, placing the spectra in this range represents the best aesthetic decision from the perspective of the author. That said, there is a slider in the GUI for the musical periodic table that allows users to change the scaling factor, thus changing the octave that the sound is in. In future work, evaluating users' assessment of (1) the pleasantness and (2) ease of audible discrimination between different elemental spectra would allow for a more empirically-based determination of a default octave for the spectra.

The decision to map the frequencies of light via a direct, linear scaling process should also be noted. This particular method considers the collection of light frequencies as a musical scale, and the scaling operation as equivalent to transposing that scale. This direct transposition process retains all of the musical intervals exactly, whereas other mappings would distort these musical relationships. Notwithstanding, alternate mapping protocols could lead to other interesting musical structures and potentially have value as an auditory data representation, and these considerations are discussed in section 4.

2.2 Sonification of Element Spectra

Each element possesses a unique atomic emission spectrum, a series of lines with discrete frequencies of light (i.e. colors) that result from electrons changing energy levels within atoms. A selection of emission spectra is included in Figure 2.

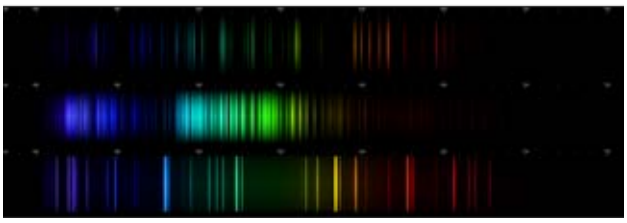


Figure 2. Examples of three element spectra. From top to bottom: oxygen (O), iron (Fe), and gold (Au).

These spectra are useful for identifying elements in chemistry and other fields, for example identifying trace heavy metals in water or discerning compositions of stars. Spectral data was obtained from an online database [14] of high-resolution emission spectra (compiled from datasets from MIT Wavelength Tables [15] and the NIST Atomic Spectrum Database [16]), in the visible range of 380-760 nm. These data were collected into TSV files, which are read in the Max/MSP patch. For each spectral line, a unique sinusoid is generated with the frequency corresponding to the frequency of the light, and the amplitude corresponding to the line's intensity. The amplitude values are normalized so that the sine wave resulting from the most intense line for each element has an amplitude of 1. As these spectra are amplitude spectra and contain no phase information, the initial phases of all sinusoids are set to 0. The sonification of helium's emission spectrum via this process is shown in musical staff notation in Figure 3.

As these spectra yield rich microtonal collections, they cannot be notated in a 12TET notation system. Cent de-

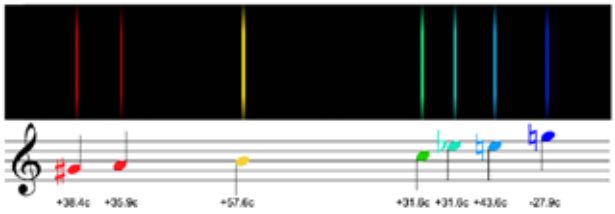


Figure 3. Sonification of helium's visible emission spectrum by scaling the frequencies of light down by 10^{-12} to produce a unique musical pitch for each color. Cent deviations are included below the notated pitch to show the actual frequencies of helium's sonification (where 100 cents = 1 semitone).

viations are included below the notes to indicate the actual frequency. The Max object ioscbank∼ is used to efficiently generate the large numbers of sine waves (which can be up to 2000 for some elements) from an array of amplitudes and frequencies. This object also enables smooth interpolation between changing frequency inputs, which allows the user to smoothly transpose the entire spectrum to a different range as discussed in section 3.

2.3 Visual Display

The visual display is generated using Jitter, with the brightness and width of each line scaled proportionally to its intensity. It is impossible to accurately represent every single spectral color with the RGB color space used in computer visuals, so an approximation method [17] was used to convert wavelengths of light into RGB values.

3. INTERACTION DESIGN

In the main window of the Max patch in presentation mode, shown in Figure 4, users can click on an element to see its spectrum and hear a sonification of it.

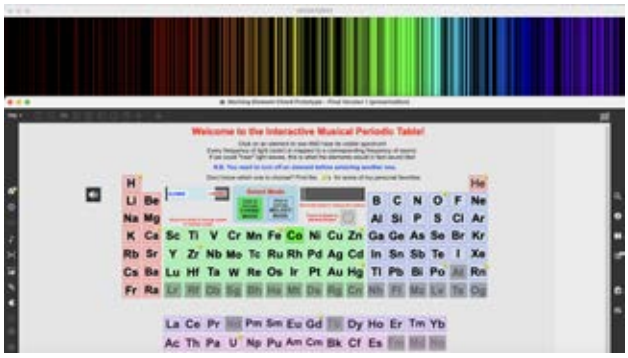


Figure 4. GUI for the Interactive Musical Periodic Table, designed in Max/MSP and Jitter.

The spectrum is displayed in a separate jit.world window, which can be displayed on the same screen, but ideally on a separate monitor or with a projector in a dark room for best effect. A user can select "CHORD MODE" to see and hear all spectral lines at once, or "MELODY MODE" to iterate through the lines one-by-one (with a short sinusoidal pulse synchronized to the appearance of each line) from violet to red (high frequency to low frequency). Within "CHORD MODE" one can also scale the octave of sound that the visible light is scaled to, and press a button to return to the default range of 380—750 Hz. Finally, within "MELODY MODE," the user can also alter the speed of playback of the lines.

The labels for the elements in the GUI are colored according to their block in the periodic table. Certain elements are greyed-out as these elements are too radioactive to measure their spectra, hence no spectral data is available and no are not sonified.

One aspect of the user design that has proved problematic is that users are required to turn off their current element selection before selecting another one. the next user need to be explicitly told this and often forget. The table only displays the visuals and audio for the last selected element, so this does not create an issue with the sound or visuals. However, this results in a number of elements remaining selected, requiring the next user to first de-select an element they would like to hear and then re-select it. Future implementations will address this issue.

4. RESULTS AND DISCUSSION

The sonifications obtained from the interactive musical periodic table vary from simple sonorities resembling tonal structures to complex noises. Simpler elements like hydrogen and helium contain only a few lines, and thus yield relatively simple combinations of tones. These simple elements can even create recognizable chords. Hydrogen has four spectral lines that form a sonority resembling a second-inversion minor chord with an added second scale degree, while helium sounds like a diatonic cluster. Users are often surprised by how closely some of the elements resemble tonal sonorities-especially that hydrogen sounds like such a simple, recognizable chord. It should be emphasized that these resemblances to tonal structures are not the result of any tempering or quantization to a chromatic or diatonic scale. Rather, hydrogen forms a microtonal pitch collection that happens to be very close to a minor chord with an added second scale degree.

While some elements–especially those towards the beginning of the periodic table with only a few electrons and therefore few spectral lines–sound like simple, tonal chords, most elements have hundreds or thousands of lines and yield much more complex, noisy clusters. Because of this, many elements become difficult to distinguish from other elements. For example, oxygen and iron sound relatively similar despite the fact that iron has approximately ten times as many spectral lines as oxygen does. This makes sense as an incredible wealth of information is being compressed into a single octave of sound. Even so, differences in elements become apparent with continued and attentive listening, just as one can learn to differentiate between different colors of noise. Additionally, the compact clusters of sine waves result in different beating patterns for the different elements, which can aid in differentiating the sounds (provided that one is able to listen for more than a few seconds).

In the author's experience, it is sometimes easier to distinguish between elements by listening to the spectral sonifications rather than looking at their visual spectra, which are also quite complex and difficult to distinguish without study. Of course, this represents a sample size of one, and the author's musical background presents a bias. Nevertheless, this suggests that this sonification method may be used to develop alternative tools for the interpretation of spectroscopy data, possibly for blind or visually impaired students or even as a supplementary tool to complement visual analysis. However, the present difficulty of distinguishing the spectra from sound suggests that a different mapping process from light frequencies to sound may be more effective in this regard–for example, one that makes use of the full extent of the human hearing range.

Visible emission spectra encode chemical properties of the elements, and there are some cases where these properties can be heard in the sonifications. For example, elements 22-30 comprise the first row of the transition metals, and include commonly known and very useful metals like iron, copper, nickel, titanium, cobalt, and zinc. All of the elements in this row, with the exception of zinc, produce dense visual spectra and correspondingly complex, noisy sonifications. However, zinc's comparatively sparse spectrum produces a simple, resonant chord reminiscent of a choir of angels singing with a delicate vibrato. Zinc's relatively simple spectrum–which can be clearly seen and also heard–is a result of its comparatively simple electron structure as compared to its other transition metal neighbors. This is just one example; further exploration of chemical properties and their relation to the sounds produced from these sonifications could provide a valuable tool for "hearing" trends in chemical properties in the periodic table. This could be useful in chemical education at the high school and collegiate level, as understanding chemical trends in the periodic table is an important but challenging part of the chemistry curriculum.

Compositionally, these sonifications provide an incredible repository of microtonal pitch collections that can be used to construct chords and melodies. This is demonstrated clearly in the author's "Helium Dance Party," where helium and other elements are used to make fun, groovy melodies. Additionally, the light-to-sound mapping can be applied outside of chemistry entirely, purely to develop colorful and eye-catching audiovisual representations of music. As the mapping of light to sound described in 2.1 associates one octave of sound with the full visible spectrum, octave equivalence can be applied such that each pitch class in a piece of music (with infinite microtonal resolution) can be represented as a single line with a unique color. One application of this is to develop synchronized light displays accompanying music for a "false synaeshtesia" experience. Such a program has been developed by the author, where users can upload a MIDI file and render a synchronized light show based on it [18].

5. CONCLUSIONS AND FUTURE DIRECTIONS

A basic prototype for sonifying the visible emission spectra of the elements was developed, and a GUI allows users to hear and see the spectra for each element. Continued refinement of the program and presentation of the tool to schools, along with surveys and assessments to gauge students' learning and interaction with the software, is in progress.

This work could potentially find application in areas spanning education, research, science communication, and composition. This work also has potential utility for spectroscopic data analysis. Sonification of spectral data is especially promising due to the fact that our ears essentially have a built-in Fourier transform; the basilar membrane in our ears deconstructs audio signals into their component sine waves. In contrast, our eyes do not possess this unique ability of spectral decomposition—that is why it is impossible for our eyes to separate out the red and yellow components from an orange paint, and why computer RGB displays can fool our eyes into thinking we see every color. We need prisms and/or computers to see the spectral components of light mixtures, but our ears pick out these spectral components automatically.

In this avenue, future work will involve experiments with focus groups to evaluate the ability of people (from musical and non-musical backgrounds) to distinguish between spectra visually and/or aurally to determine if sonification of visible element emission spectra is as effective as visualization of the spectra.

6. VIDEO EXAMPLE

A short video demonstrating the interactive musical periodic table in action, with examples of the element sonifications, can be found at the following link:
`https://drive.google.com/file/d/1Vip9FK3LUlwAOh74fPBf_WBMF14fMBTZ/view?usp=sharing`

Acknowledgments

This research received funding support from several Indiana University departments, including the Undergraduate Research Council, the Center for Rural Engagement, the Jacobs School of Music, the Jonson Center for Entrepreneurship & Innovation at the Kelley School of Business, and the Hutton Honors College. Additional funding was provided through a SEAMUS CREATE Grant.

7. REFERENCES

[1] T. Delatour, "Molecular Music: The Acoustic Conversion of Molecular Vibrational Spectra," *Computer Music Journal*, vol. 24, p. 48–6, 2000, https://direct.mit.edu/comj/article-abstract/24/3/48/93441/Molecular-Music-The-Acoustic-Conversion-of?redirectedFrom=fulltext.

[2] F. Pereira, J. C. P. e Sousa, R. P. S. Fartaria, V. D. B. Bonifácio, P. Mata, J. A. de Sousa, and A. M. Lobo, "Sonified Infrared Spectra and Their Interpretation by Blind and Visually Impaired Students," *J. Chem. Educ.*, vol. 90, no. 8, pp. 1028–1031, 2013, https://pubs.acs.org/doi/10.1021/ed4000124.

[3] S. Munukutla, A. Bertoy, S. R. amd, and A. Ramamoorthy, "Molecular Music: A Modern Accompaniment to NMR Pedagogy," *J. Chem. Educ.*, vol. 99, no. 2, pp. 810–818, January 6 2022. [Online]. Available: https://doi.org/10.1021/acs.jchemed.1c01097

[4] I. P. Varela, G. Shear, and C. Cobas, "Molecular Melodies: Unraveling the Hidden Harmonies of NMR Spectroscopy," *Molecules*, vol. 29, no. 4, p. 762, Feb 2024, https://www.mdpi.com/1420-3049/29/4/762.

[5] F. Morawitz, "Molecular Sonification Of Nuclear Magnetic Resonance Data As A Novel Tool For Sound Creation," 2016, pp. 6–11, https://quod.lib.umich.edu/cgi/p/pod/dod-idx/molecular-sonification-of-nuclear-magnetic-resonance\protect\penalty\z@fdata-as.pdf?c=icmc;idno=bbp2372.2016.002.

[6] W. Smith, "The Sound of Molecules," https://www.youtube.com/watch?v=b3e7W1ak2BQ, 2021.

[7] D. Harvey, "Atomic Emission Spectroscopy," https://chem.libretexts.org/Bookshelves/Analytical_Chemistry/Analytical_Chemistry_2.1_(Harvey)/10%3A_Spectroscopic_Methods/10.07%3A_Atomic_Emission_Spectroscopy, May 2024, accessed: 20 May 2024.

[8] W. Smith, "Chromatic Chemistry: the Periodic Table in Light and Sound," https://www.youtube.com/watch?v=Z9dpHWrgzMw, 2022, 8:22 – 14:16.

[9] C. Shakalis, "Mixing music and science: IU student shares 'The Sound of Molecules' at WonderLab," *The Herald-Times*, 2022.

[10] O. Darrigol, "The analogy between light and sound in the history of optics from the ancient greeks to isaac newton. part 2†," 2010, vol. 52, no. 3, pp. 206–257.

[11] I. Newton, *Opticks: or, A Treatise of the Reflections, Refractions, Inflexions and Colours of Light*. London: Printed for Sam. Smith, and Benj. Walford, 1704, retrieved from https://doi.org/10.5479/sil.302475.39088000644674.

[12] T. Young, "The Bakerian Lecture. On the theory of light and colours," 1802, https://doi.org/10.1098/rstl.1802.0004.

[13] E. A. Lopez-Poveda, *Development of Fundamental Aspects of Human Auditory Perception*. Academic Press (Elsevier), 2014.

[14] J. G., "Visible spectra of the elements," https://atomic-spectra.net/index.php?lang=en.

[15] F. M. P. III, *MIT Wavelength Tables, Second Edition*. MIT Press, November 12 1982.

[16] Atomic Spectra Database, "NIST," https://www.nist.gov/pml/atomic-spectra-database.

[17] D. Bruton, "Approximate RGB values for Visible Wavelengths," SFA Physics, 1996, Note: The original URL appears to be inactive as of June 2024, but a copy of the code is available on https://www.alanzucconi.com/2017/07/15/improving-the-rainbow/, under the section 'Bruton Colour Scheme.".

[18] W. Smith, "Element Visible Emission Spectra Sonification," https://colab.research.google.com/drive/166S2Pl0ZN_N5qK19YlF__aQurmowr-Ki.

INTERPRETATIVE DATA SONIFICATION: USING LLMS TO INTERPRET DATA AND GENERATE CONTINUOUS SOUNDSCAPES AT THE SYDNEY OPERA HOUSE

Rodolfo OCAMPO[1], **Oliver BOWN**[1], **Brendan WRIGHT**[2,1], **Justin SHAVE**[2], and **Caroline PEGRAM**[2]

[1]**University of New South Wales**, Sydney, Australia
[2]**Uncanny Valley Music**, Sydney, Australia

ABSTRACT

This paper describes a generative music work, 'Music of the Sails' developed as a commission for the 50th anniversary of the Sydney Opera House. The artwork consisted of a generative composition that responded to the building's data continuously in real-time for one month. The GPT-4 large language model (LLM) was used as an interpretative interface between the live data and a generative music engine. The data included information from the building's climate and energy systems, such as temperature and energy use, and its roster of concerts. The composition consisted of a generative arrangement of readymade music assets, including field recordings, AI-generated music, and human-composed music. Our approach, called Interpretative Data Sonification, leverages LLMs to interpret the building's data and use it to drive a generative soundscape. Music of the Sails was live-streamed on the Sydney Opera House website continuously for 30 days (744 hours) and exhibited as an audiovisual installation over three days. This research explores the roles that generative artificial intelligence (AI) can play in the interpretation of data as sound. We describe challenges faced around LLM output fixation and primacy bias and how future work can address these.

1. INTRODUCTION

This paper describes a generative music work, Music of the Sails, commissioned by the Sydney Opera House for its 50th anniversary. The brief was to use AI to bring the building to life, to reveal it not just as a place of performance but as a performer itself. In particular, they wanted to showcase the systems that inhabit the building, especially its custom seawater cooling system. To achieve this, we had access to near real-time data from systems that regulate the temperature of the building, air quality, water, energy and the monthly schedule of performances across the six venues inside the Opera House.

Building on previous work [1] we developed an approach combining data sonification, soundscape design and generative AI. Data sonification transforms data into sound to convey information sonically for various purposes, including data exploration, entertainment, and artistic expression [2, 3]. It often involves mapping numerical or categorical values to musical qualities such as tempo, pitch, or key [4]. However, these approaches often face the challenge of meaningfully conveying the qualities of the data sonically [5].

In contrast, human musicians transpose complex phenomena into musical compositions with layers of meaning. Classical approaches to the musical representation of the environment are interpretive, not direct. A classic example is Vivaldi's Four Seasons, capturing nuanced qualities of the changing seasons through music [6]. We sought to develop an approach that both reacted to data but that could integrate elements of human compositional skill and creative interpretation to richly convey the underlying phenomena behind the data. We used the capability of LLMs to manipulate simple concepts to interpret the data and then drive a music generation engine designed and curated by a human composer.

In a previous installation [1], we developed a first approximation of this approach that generated soundscapes by matching data descriptions generated by LLMs to a set of human-created audio descriptor tags using word embeddings and cosine similarity. The current implementation evolves that approach. Instead of performing latent space matchings of word embeddings of data and audio labels, we leverage language models to not only generate interpretations of data, but also to make selections about how the soundscape generation engine will work. In this way, we engage the LLM's capabilities more deeply and employ it as a tool with greater potential creative agency. The LLM was primed with information about its role and the artistic goals of the installation, as well as context about the data, the Sydney Opera House, and the music generation engine, providing the basis for it to make decisions about what music elements to select.

Candy [7] suggests AI can take multiple creative roles, including tools, mediums, mediators and partners. As a tool, the system is used by a human for a specific purpose and its effectiveness relies upon human skill. As a mediator, the system enables complex relationships between people, technology and the environment. As a partner, the system acts with a degree of agency, influencing the creative output non-deterministically.

In our installation, the AI seemingly expresses elements of each of these roles. It served as a tool to generate a human readable interpretation of data, as a mediator between the complex systems inhabited by and inhabiting in

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the building, and as a partner that made selections about the creative output experienced by the audience. However, in each of the roles, we encountered limitations, particularly in the tendency to fixate around certain outputs and the exhibition of primacy bias. In this paper, we discuss these limitations and possible avenues to address them.

2. RELATED WORK

The direct creation of soundscapes or music from data has been explored in a variety of contexts, from scientific exploration to artistic expression. Artists such as Marty Quinn's [8] and Andrea Polli [9] have explored sonification of data from various scientific fields, including ice core, radar, DNA, seismic, and solar wind data, with both aesthetic and data analysis goals.

More recently, several approaches have explored Natural Language Processing approaches to perform mappings between language and music. For example, Krol et al. [10] explored the use of GPT-3 to generate musical explanations, showcasing the potential of LLMs in creative musical contexts. Successful text-to-music generation has rapidly emerged in recent years. Companies like Mubert [1] combine user prompts, latent embeddings, and human-composed sound clips to generate music. This approach shares similarities with our work in terms of using semantic matching between text and sound.

In this context, Pigrem et al. [11] proposed that data sonificiation and soundscaping are two sonic art forms that have developed partly in isolation but that share important similarities, and thus proposed the concept of datascaping, where data sonification serves as a narrative device in soundscape composition. Their approach uses recorded sounds augmented with real-time sonified elements. Our approach is similar to datascaping, where our goal is to let data elements drive a rich, human-composed sonic experience. However, our approach also integrates the use of LLMs as a bridge between the two.

In our previous work [1], we developed Semantic Sonification, which involves a more direct mapping between data descriptions and selected audio elements. In the current approach, which we term Interpretative Data Sonification, we replace cosine similarity matchings of multimodal word embedding tags with an LLM that can drive the soundscape generation process based on an interpretation of the data, the soundscape engine and our artistic goals. We also involve a human compositional process more deeply, creating many fixed elements for the soundscape, along with AI-generated music loops, and field recordings inside the Opera House building.

3. IMPLEMENTATION

3.1 The Opera House Systems

The Opera House is renowned for its sophisticated self-regulation and energy efficiency systems [12]. It can be understood as an adaptive system that seeks to maintain homeostasis across multiple variables, including temperature, air quality, energy use, and water use. To achieve this, it employs a complex set of interconnected monitors and controls.

For example, the building uses seawater from the harbour in its cooling system. The water is circulated to cool the building and then returned to the harbour at a temperature that does not disrupt the ecosystem. This cooling system is powered by a set of six chillers that are activated in stages based on the cooling needs of the building. Like other buildings it also regulates CO2 levels to maintain acceptable air quality. It draws in outside air to reduce CO2 exhaled by people.

The Opera House contains several venues, including the Concert Hall, the Utzon Room, the Drama Theatre, the Playhouse, and the Studio. When these spaces fill up, such as during performances, the CO2 levels and temperature can rise quickly. In anticipation of heating, the building preemptively cools the venues before performances begin. To compensate for increased CO2 the building's systems draw in more outside air, which makes cooling less efficient than using recycled air. Consequently, the building is always in a state of dynamic adaptation, responding to both internal and external conditions.

3.2 The installation

Our installation needed to reflect the dynamic, continuous nature of the building's adaptive systems. It connected to these systems in real-time, generating a continuous soundscape that played 24 hours a day for 30 days over a livestream and as part of a 3-day audiovisual installation at the TIDE room inside the Sydney Opera House.

The building moves through different states throughout the day, and our goal was to generate music that reflected each of these states. For example, when the building was hot and busy, we wanted the soundscape to be lively, full of sounds, and to convey a sense of warmth. Conversely, during quiet periods in the middle of the night with no performances, we wanted the soundscape to feel calm and cool.

3.3 Interpretative Data Sonification

We received the data from the building each hour in the form of a CSV file containing minute-by-minute data points. These data were preprocessed to a simplified summary, formatted as a string and passed to the LLM in a prompt along with context about the meaning of variable names and ranges.

We then asked the LLM to generate an interpretation of the data that succinctly described what it said about the state of the building. We then gave the LLM information about the Memu music generation engine (see below), and asked it to select parameters controlling its state.

3.3.1 The Memu Music Generation Engine

Memu is a music generation system that combines ready-made stems with rule-based manipulation. As part of our installation, we utilized a variety of sound sources, including custom compositions by the fourth author, recordings

[1] www.mubert.com

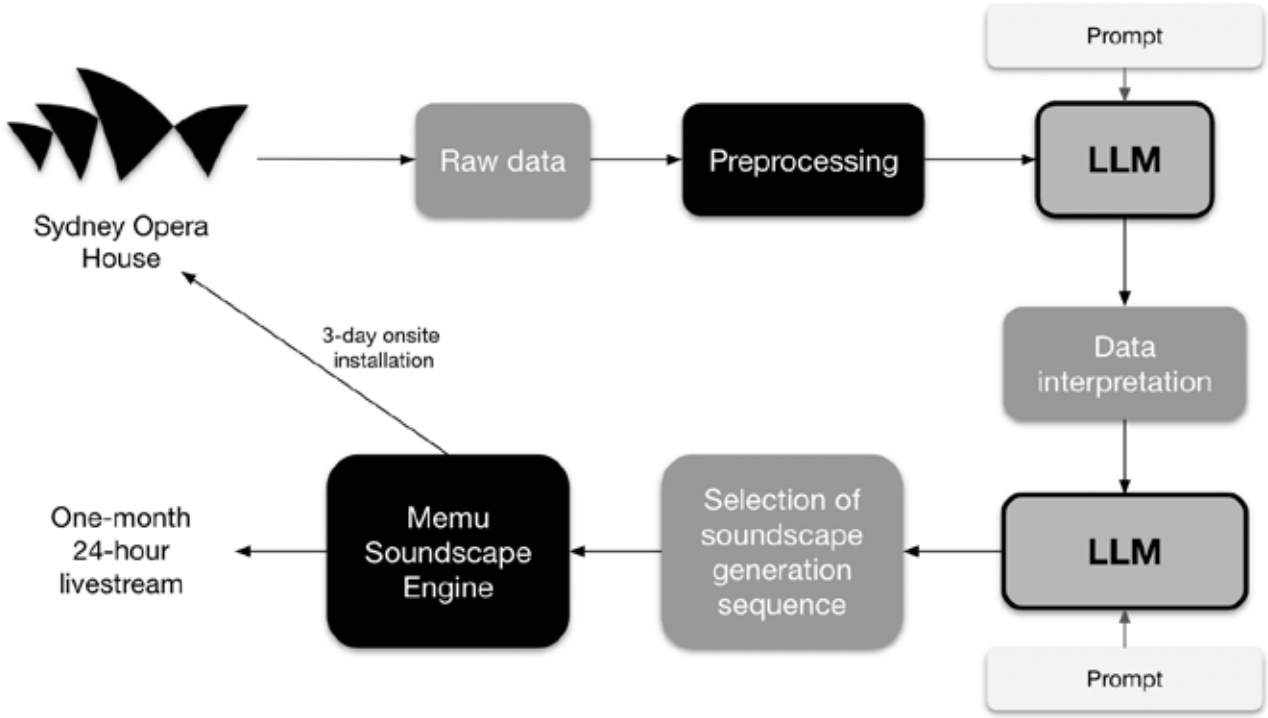


Figure 1. Flow of information for the installation, from data ingestion to soundscape generation.

Figure 2. The recordings from the Sydney Opera House building included sounds from the working infrastructure, which became part of the generative soundscape

from the Sydney Opera House's Grand Organ, field recordings from around the building, such as the chillers, water pumps, and electrical systems, as well as variations of these sounds generated by AI processes.

3.3.2 Sequences

Memu generates music in different modes of operation, termed *sequences*. Each sequence involves a set of combinatorial rules that seek to evoke a particular emotion, producing never-repeating 5-minute tracks that get interwoven to produce a continuous soundscape.

Each sequence was extensively labeled by the composer-engineer with the feeling it aimed to convey. Table 1 shows six examples out of the 22 crafted sequences.

The LLM was asked to interpret the data from the building and then select three sequences that would match the building's state, from a list of all possible sequences. This was done once per hour for 744 hours.

3.3.3 Data Collection and Preprocessing

The data from the Sydney Opera House consisted of minute-by-minute raw data from each room's regulating systems, including CO2 levels, room temperature, water temperature, ongoing events, as well as energy consumption from the building's chillers.

Table 2 shows an example of part of the data.

These data were formatted as JSON and passed hourly to the LLM as part of the prompt string.

3.3.4 Data Interpretation

We asked the LLM to generate a succinct description that captured the state of the building based on the incoming data. For example, for the data in table 2, the following description was generated.

Generated text:

> *Late evening at the House finds three chillers powering the cooling system, keeping the Drama Theatre and the Playhouse primed for imminent performances. The Joan Sutherland Theatre and other venues remain in a state of*

Table 1. Sequence labels. Sequences are series of rules that generate unique soundscapes. The labels are human-written descriptions of the sequences.

No.	Description
1	The music exudes feelings of joy and optimism. The harmonious melodic fragments invoke a sense of contentment and happiness, creating a serene and uplifting mood that resonates with feelings of hope and positivity.
2	The musical sequence evolves from serenity to vibrancy, with a shift in tonal dynamics and rhythmic intensity. It encapsulates the subtle transition from tranquility to lively anticipation.
3	In this sequence we reach an unwavering stability, offering a rhythmic continuity that balances sense of tranquil constancy and fervent activity.
11	Harmonious layers intertwine in a relentless, dynamic equilibrium, generating an atmosphere of thoughtful, energized tranquility.
21	The rhythmic experience unfolds in slow, disjointed cadences, each beat a spectral echo weaving through a dimension of languid resonance and ephemeral pulses.

tranquility, their temperatures and CO2 levels stable under a watchful system.

3.3.5 Soundscape Generation

Based on that description, we asked the model to select three sequences of operation for the Memu generative music system that best accommodated the building's state.

We also asked to justify how these soundscapes were suitable, and the justification was shown on screen as part of the visual elements of the installation.

Generated text:

Inspired by these data, the rendered soundscape encompasses the vibrancy of the theatre performances and the tranquility of the building's operation. The harmonious melodies reflect the joy of the shows, the transition to vibrancy mirrors the anticipation of the audience, while the constant rhythmic elements mirror the building's stable operation.

Selected soundscapes ids: 1, 2, 3

3.4 Audiovisual Installation and Live Stream

Visual elements were also included as part of the soundscape generation. The visuals were created using the generative video system Deforum, and involved five-minute clips of merging visuals that include motifs from the Sydney Opera House, themes from the event calendar, and different moods and themes corresponding to the music labels.

Table 2. Selection of Opera House data elements.

Data Point	Value
Active Chillers	Ch1, Ch2, Ch5
Chiller Stage	Stage3
Chiller Load (Ch1)	1112.883 kW
Chiller Load (Ch2)	964.2559 kW
Temp. (Concert Hall)	22.16°C
Temp. (Opera Theatre)	22.25°C
Water Temp. (Ch1)	15°C
Water Temp. (Ch2)	16°C
CO2 (Concert Hall)	451.4 ppm
CO2 level (Concert Hall)	Normal
CO2 (Opera Theatre)	406.8 ppm
Valve (Concert Hall Cold Water)	Stable
Valve (Opera Theatre Hot Water)	Stable
Temp. Oscillation (Concert Hall)	Stable
Event (Playhouse)	Twelfth Night

Overlaid on these visuals, the outputs from the LLM were displayed at regular intervals. These include the data interpretation and a musical explanation of how the generated soundscape reflected the data qualities. Lastly, graphs of the data were also displayed at regular intervals.

Figure 3 shows stills from the live stream displaying the generated text interpreting the data, and the data history for different venues, which appeared on screen at regular intervals.

When text or data was not being displayed, a stream of generative video visuals was shown, as shown in Figure 4.

4. DISCUSSION

A total of 744 hours of live data from the Sydney Opera House were used to drive the generation of music, live-streamed online on the Sydney Opera House website, and displayed as part of an audiovisual installation at the TIDE Room for three days, inside the House. To hear the soundscape and see the data and interpretations generated by the LLM model, the reader can access a 24-hour archive and a highlights reel from the entire month [2] .

The work demonstrated how an LLM could perform interpretation between a live data stream and a generative musical work. The use of an LLM as an interpreter or mediator afforded a different approach to data soundscape creation which was not constrained to establishing direct mappings from data to sound parameters. Instead, it enabled us to work in a more traditional mode of creative interpretation, freely developing musical material that reflected our impressions of the building and its various states and activities, and then orchestrating the ways in which the LLM selected this content. The design of prompts, including information about the building data, the set of music possibilities to choose from, and the creative objectives, enabled us to control the context that informed how the LLM interpreted the data. However, in practice the work encountered the natural limitations of what cur-

[2] https://stream.sydneyoperahouse.com/music-of-the-sails

Figure 3. (Left) Text generated by the LLM interpreting the data and explaining the generated soundscape. (Right) Data visualization displayed during the installation, showing real-time data from the Sydney Opera House.

Figure 4. A still from the generative audiovisual stream

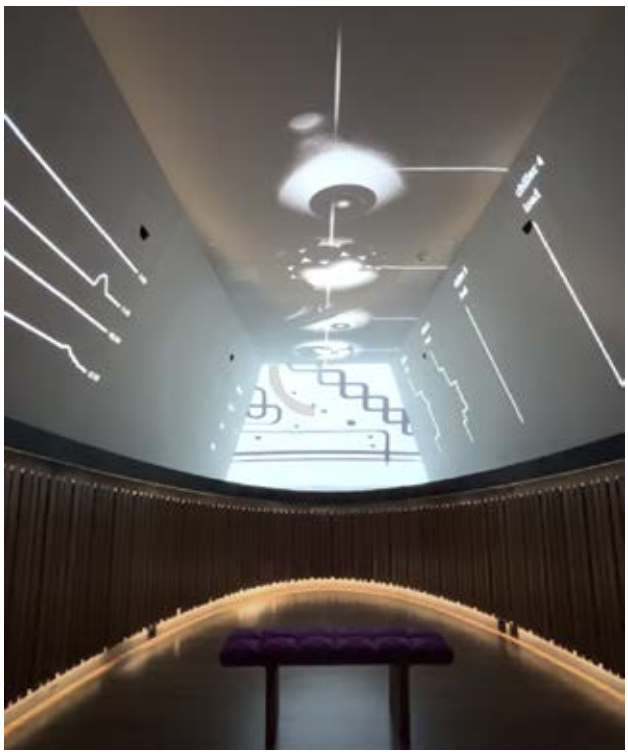


Figure 5. Photo of the onsite installation inside the TIDE room at the Sydney Opera House

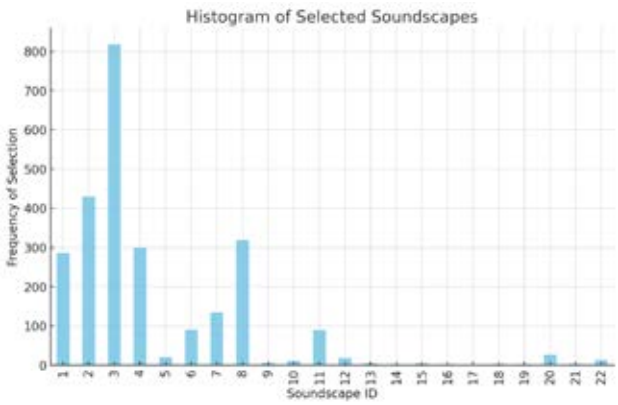


Figure 6. Histogram displaying sequence selection frequency

rent LLMs can do given this kind of task.

What was particularly exciting about this approach was the idea that, in principle, one could then maintain an ongoing dialogue with the LLM as it processed the data. Thus over the 744 hour performance it could be possible to give feedback and additional creative input. This was not practically possible in the present work due to processor constraints.

4.1 Limitations and Future Research

We identified limitations in our current approach. One of the objectives of our installation was to produce variety in soundscape generation. However, after examining the record of the selected soundscape sequences by the LLM, we found that it tended to fixate on certain sequences.

Figure 6 shows a histogram of the selected sequences for the entire month. As we can observe, the model tended to focus on a narrow number of sequences.

This can be due to several reasons.

4.1.1 Lack of Data Variety

The Sydney Opera House is a well-regulated building that maintains a steady equilibrium. Therefore, the model tended to select soundscape sequences that were tagged with words that reflected this stability, as well as positive terms that reflected excitement for events or coming down

from activity. In contrast, it did not select sequences for soundscapes tagged with words that reflected instability, or negatively-valenced terms.

Table 3 shows the tags for the most frequently selected sequences.

Table 3. The most frequently selected sequences for soundscape generation

No.	Description
1	The music exudes feelings of joy and optimism. The harmonious melodic fragments invoke a sense of contentment and happiness, creating a serene and uplifting mood that resonates with feelings of hope and positivity.
2	The musical sequence evolves from serenity to vibrancy, with a shift in tonal dynamics and rhythmic intensity. It encapsulates the subtle transition from tranquility to lively anticipation.
3	In this sequence we reach an unwavering stability, offering a rhythmic continuity that balances sense of tranquil constancy and fervent activity.
4	This sequence gradually transitions into serenity as its rhythmic complexity diminishes, leaving behind a lingering sense of calmness and grace.
8	This music provides a soothing backdrop, while descending melodies evoke a sense of tranquil descent. This musical passage carries an air of introspection and peaceful reflection, offering a harmonious journey back to a state of quietude and contentment.

Table 4 shows the tags for the sequences that were not selected, which include more words referring to tension, disquiet and melancholy.

Figure 7 shows a wordcloud with size weighted by frequency of all the descriptions generated by the LLM in response to the Opera House data. As we can see, these words align more closely with the descriptions shown in table 3, corresponding to the most frequently selected sequences.

4.1.2 Primacy Bias

However, another factor that might have influenced the fixation was a well-documented bias in large language models: primacy effect or primacy bias. Several authors have reported this bias on large language models, where they tend to select the first elements in a list [13, 14]. This primacy effect is similar to primacy biases exhibited by humans and it is currently an active area of research since many of the applications of LLM's involve selecting elements from lists in semantic-based tasks. Primacy bias may have significantly influenced how the LLM selected options from our list of music possibilities.

Table 4. Sequences that were not selected at all by the system

No.	Description
14	This sequence, while ethereal and luminous, emits waves of unsettling and disconcerting harmonies, casting a celestial, yet uneasy resonance through the escalating rhythmic landscape.
15	The music pulses in a dynamic equilibrium, its harmonies interlaced with intricate, relentless patterns, creating an otherworldly tapestry that is both luminous and subtly disquieting, creating a dance balance between comfort and unease.
16	The rhythms start to unwind, soothing the previous tension and disquiet, allowing the ethereal harmonies to envelop the soundscape in a calming, albeit still subtly unsettling, embrace.
17	This sequence piece unfolds like a haunting, introspective whisper, weaving through the shadows with its melancholic harmonies and enveloping atmospheres.
18	This sequence commences a swell into reflective majesty, intensifying melancholic echoes and enveloping the senses in a crescendo of solemn magnificence.
19	The harmonies resonate within a majestic equilibrium, each somber note hovering in a monumental rhythmic dance of enduring resonance, creating feelings of solemn grandeur.
21	The rhythmic experience unfolds in slow, disjointed cadences, each beat a spectral echo weaving through a dimension of languid resonance and ephemeral pulses.

4.1.3 Dialogic Interaction to Address these Limitations

We believe there are two main ways to address the fixation on sequence selection in our Interpretative Data Sonification pipeline. Feedbacking the history of its previous selections (history awareness) and engaging in a dialogue in real-time to steer the LLM's behaviour.

4.1.4 History Awareness:

Feedbacking the history of its previous selections to the model, along with a specific prompted goal of avoiding repetitiveness, we believe we could partly address the issue of fixation and primacy effects. In this installation, we did not include such previous history to save inference cost and time. However, with increasing context token windows and the ability to include moving windows that progressively summarise inference history, including large amounts of previous context becomes more feasible.

4.1.5 Dialogic Interaction:

On the other hand, being able to engage in a dialogue with the LLM and say things like: "you have selected sequence three too many times, try selecting others" could more eas-

Figure 7. Word cloud of the descriptions generated by the LLM in response to the Opera House data, with size weighted by frequency.

ily help address the fixation. During the installation, we partly modified the prompts in real-time to try to steer the model in a particular direction after observing undesired behaviour. A step beyond this would include a chat interface where the human creator would engage in a running dialogue with the LLM performing the Interpretative Data Sonification matching and therefore be steered more effectively without prompt trial and error. This, combined with history awareness, both of previous selections and conversations with the user, would provide better adherence to diversity goals.

Bown et al. [15] explore the concept of dialogic interaction as a mechanism to enable more effective human-AI co-creative interactions. We believe future installations would benefit from more explicitly enabling dialogic forms of collaboration with generative AI models.

Such a dialogue would elevate the AI model even more to a co-creative and active role, in which it is able to adjust to changing objectives and even be flexible with regard to the initially established creative objectives.

5. CONCLUSIONS

In this paper, we presented 'Music of the Sails', a generative soundscape installation developed for the 50th anniversary of the Sydney Opera House that combines data sonification, human soundscape design, and generative AI to create a sonic celebration of the House's changing states. Our approach, termed Interpretative Data Sonification, leverages large language models (LLMs) to interpret the building's real-time data and drive a generative soundscape engine. The project explored the roles of LLMs as creative tools in sonic installation, in which human compositional skill serves as the backbone of a generative data-driven process.

The experimental work revealed limitations of this method, particularly the LLM's tendency to fixate on certain musical sequences, possibly due to primacy bias and limited data variability, which constrain the variety of soundscapes. To address these issues, we propose feedbacking the history of the model's previous selections to encourage diversity and enable real-time dialogic interaction for dynamic adjustments to the model's behavior. Despite the outlined challenges, 'Music of the Sails' contributes to the growing body of research on AI-driven creative systems, highlights the potential of LLMs as realtime interpreters or mediators between complex data and artistic expression, and opens up new modes of interaction between human artistry, generative AI, data and sonic art.

6. ACKNOWLEDGMENTS

We would like to thank the Sydney Opera House team: Stuart Buchanan, Melinda Dransfield, Rob Millet, and James Archer.

We also thank the entire Music of the Sails team: Chloe McFadden (generative visuals), Charlton Hill (management), Stephen Krol (AI-music stems) and Felix Wallis (additional music).

We acknowledge the Gadigal people, the traditional custodians of the land where the Sydney Opera House stands.

7. REFERENCES

[1] R. Ocampo, J. Andres, A. Schmidt, C. Pegram, J. Shave, C. Hill, B. Wright, and O. Bown, "Using gpt-3 to achieve semantically relevant data sonificiation for an art installation," in *Artificial Intelligence in Music, Sound, Art and Design: 12th International Conference, EvoMUSART 2023, Held as Part of EvoStar 2023, Brno, Czech Republic, April 12–14, 2023, Proceedings*. Springer, 2023, pp. 212–227.

[2] D. Rocchesso, S. Serafin, F. Behrendt, N. Bernardini, R. Bresin, G. Eckel, K. Franinović, T. Hermann, S. Pauletto, P. Susini *et al.*, "Sonic interaction design: sound, information and experience," in *CHI'08 extended abstracts on Human factors in computing systems*, 2008, pp. 3969–3972.

[3] T. Hermann, A. Hunt, and J. G. Neuhoff, *The sonification handbook*. Logos Publishing House Berlin, 2011.

[4] F. Grond and J. Berger, "Parameter mapping sonification," in *The Sonification Handbook*. Logos Publishing House Berlin, 2011, ch. 15, pp. 363–397.

[5] S. Barrass, "A perceptual framework for the auditory display of scientific data," *ACM Transactions on Applied Perception (TAP)*, vol. 2, no. 4, pp. 389–402, 2005.

[6] M. Talbot, *The Vivaldi Compendium*. Boydell Press, 2011.

[7] L. Candy, "Creating with the digital: Tool, medium, mediator, partner," in *Interactivity, Game Creation, Design, Learning, and Innovation*. Springer International Publishing, 2020, pp. 13–28.

[8] M. Quinn, "The climate symphony and other sonifications of ice core, radar, DNA, seismic and solar wind data," in *Proceedings of the 2001 International Conference on Auditory Display, Helsinki Finnland*, 2001.

[9] A. Polli, "Atmospherics/Weatherworks: A Multi-Channel storm sonification project," *International Conference of Auditory Display*, Jul. 2004.

[10] S. J. Krol, M. T. Llano, and J. McCormack, "Towards the generation of musical explanations with gpt-3," in *Artificial Intelligence in Music, Sound, Art and Design: 11th International Conference, EvoMUSART 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings*. Springer, 2022, pp. 131–147.

[11] J. Pigrem and M. Barthet, "Datascaping: Data sonification as a narrative device in soundscape composition," in *Proceedings of the 12th International Audio Mostly Conference on Augmented and Participatory Sound and Music Experiences*. New York, NY, USA: ACM, Aug. 2017.

[12] Sydney Opera House. (2023) Sustainability. [Online]. Available: https://www.sydneyoperahouse.com/about-us/in-the-community/environmental-sustainability

[13] J. E. Eicher and R. Irgolič, "Compensatory biases under cognitive load: Reducing selection bias in large language models," *arXiv preprint arXiv:2402.01740*, 2024.

[14] Y. Wang, Y. Cai, M. Chen, Y. Liang, and B. Hooi, "Primacy effect of ChatGPT," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 108–115.

[15] O. Bown, K. Grace, L. Bray, and D. Ventura, "A speculative exploration of the role of dialogue in Human-ComputerCo-creation," *ICCC*, 2020.

ARPEGGIATORUM: AN AUDIO CONTROLLABLE MIDI ARPEGGIATOR

Axel BERNDT (axel.berndt@uni-paderborn.de) [1] and
Davide Andrea MAURO (davide.mauro@uni-paderborn.de) (0000-0001-8437-4517) [1]

[1] *KreativInstitut.OWL*, **Paderborn University**, Detmold, Germany

ABSTRACT

An arpeggiator is a system that takes in input notes (or chords) and breaks them up into sequences, so-called arpeggios. While typical MIDI arpeggiators receive and produce MIDI messages *Arpeggiatorum* allows for an audio signal to be the input of the system, performing mono- or poly-phonic pitch estimation. *Arpeggiatorum* is also specifically designed to work with pipe organs. In this contribution we present the architecture of *Arpeggiatorum*, highlighting the design choices that lead to the development of specific features, and its strengths and current limitations. We will focus our attention on the possibility to use an audio input, for example a voice, or multiple voices, to control the system, and the new musical possibilities that emerge. We discuss the impact that of some of the most crucial parameters, namely accuracy and delay of the audio input have on the performances and suitability of the system.

1. INTRODUCTION

Modern pipe organs no longer have exclusively mechanical actions, but are more and more often equipped with electromechanical actions. This allows organ consoles to be placed more freely and be equipped with a MIDI interface to control keys, stops, swells etc. However, these organs offer barely more new features than the possibility to record and replay performances, define mixtures, transpose and route keyboards to certain works. As part of a larger activity we began to evaluate the musical and technical possibilities of modern organs that lie untapped. These unfold their full potential when a computer (or more generally speaking, a programmable MIDI device) is connected between the console and the organ. Thereby, the computer can act as a mapping device for input from the console (effectively an ordinary MIDI input device), translate, enrich or react on it. Other input devices can also be utilized to play the organ. The computer can even become an input device in itself.

One of the mechanisms that we have specifically developed to expand the playing possibilities is an arpeggiator, called *Arpeggiatorum*. It came as a surprise that, although it would have been technically possible for decades and well-proven in the world of synthesizers (see for example [1]), arpeggiators so far never made it onto the organ. In contrast to many other software and hardware arpeggiators available *Arpeggiatorum* is conceived specifically for organs. This is reflected by some of its unique features and design decisions discussed here. For a review of sequencers, a class of systems that include arpeggiators, see [2]. For a discussion on the possibilities of an Arpeggiator as a compositional tool, see [3].

The paper is structured as follows: in Section 2 we provide an overview of the system. In Sections 3 and 4 we present the implementation details, and discuss the current limitations. Finally, in Section 5 we highlight the future steps for *Arpeggiatorum* and possible directions for this research.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. AN OVERVIEW OF *ARPEGGIATORUM*

Arpeggiatorum is a Java application that implements a MIDI software arpeggiator, and it can also be controlled with audio as its input. It is released Open Source (with the GPL-3.0 license) utilizing the following libraries: JSyn [4], Tarsos DSP [5], and (optional) PortAudio [6].

The main GUI for *Arpeggiatorum* is presented in Figure 1. Three different tabs provide access to settings for MIDI (e.g. Input and Output Channel), Audio (e.g. Device, Channel, Pitch Tracking algorithm) and Arpeggiator (e.g. Enrichment Pattern, Tempo).

One of the most common scenarios for the use of *Arpeggiatorum* involves a musician performing on a MIDI equipped organ while controlling the software. For this reason a special performance mode GUI has been proposed. All relevant information should be readily available and easy to reach in a single screen and a touch-compatible interface (see Figure 2).

Settings are automatically saved and loaded from a text file upon opening and they include the Sample Rate (default 44100), parameters for pitch detection algorithms, and the state of the GUI.

3. IMPLEMENTATION

An arpeggiator's basic functionality involves an input device providing pitches (in this case *noteOn* and *noteOff* messages from a MIDI device) and the arpeggiator logic (see Figure 3). Since MIDI input devices are not necessarily limited to classic keyboards and organ consoles we were also striving for alternative input modalities. Singing is particularly relevant for church music, thus we have also added an audio input to *Arpeggiatorum* that determines

Figure 1. The GUI for *Arpeggiatorum*.

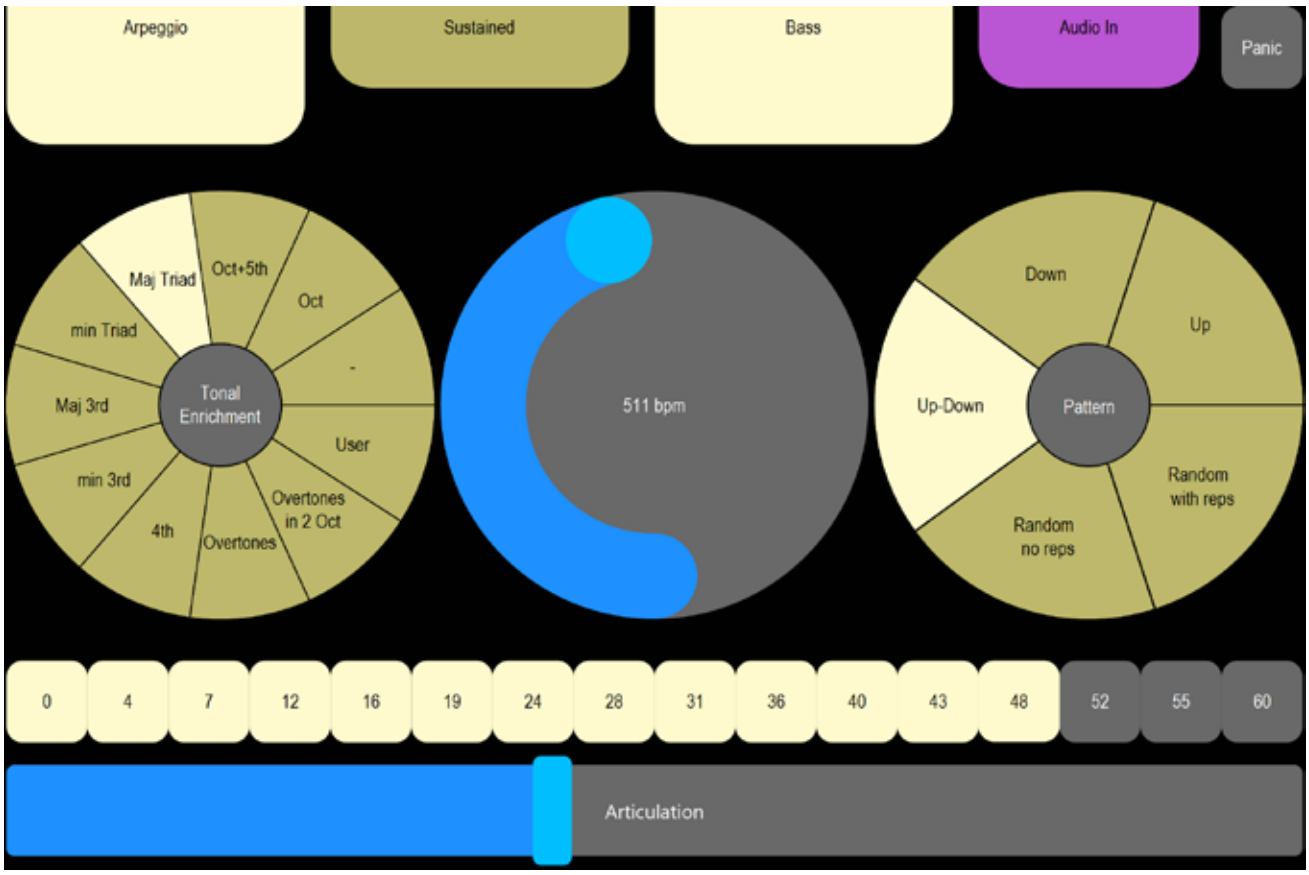


Figure 2. The GUI for Performance Mode.

sung pitches described in Section 3.1. The arpeggiator logic then enriches the pitch pool and plays them in sequences according to certain patterns, as detailed in Section 3.2.

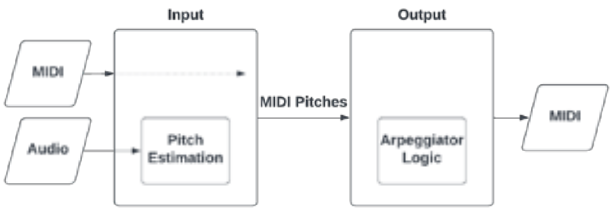


Figure 3. Functional diagram for *Arpeggiatorum*.

3.1 Audio Pitch Tracking

A novel feature of *Arpeggiatorum* is the possibility to use an audio input signal to control the arpeggiator. Various different algorithms are currently implemented, with the possibility to easily add other ones in the future thanks to the modular architecture of JSyn. In particular two classes of algorithms are currently implemented: monophonic and polyphonic pitch trackers. Monophonic pitch algorithms currently include: 1. Autocorrelation Function 2. Harmonic Product Spectrum and Cepstrum 3. FFT-YIN. Polyphonic pitch algorithms currently include: 1. Constant Q Transform.

Table 1 summarizes the lowest tracked frequency and delay introduced by each proposed algorithm.

Algorithm	Min Freq (Hz)	Delay (ms)	FFT Size
JSyn ACF	40	P + 8 samples	-
HPS/Cepstrum	86.13	12	512
FFT-YIN (Tarsos)	86.13	12	512
CQT (Tarsos)	41.21	743	32768
	82.41	372	16384
	164.82	186	8192
	329.64	93	4096
	659.28	46	2048
	1318.56	23	1024

Table 1. Lowest frequency, delay, and FFT size (where relevant) for various algorithms. P is the period of the target frequency. 8 samples is the buffer size of the internal scheduler of JSyn.

3.1.1 JSyn ACF (Monophonic)

JSyn includes a pitch detector that estimates the fundamental frequency of a monophonic signal. It utilizes the autocorrelation function and outputs the frequency as well as a confidence value. A new output is generated at every update of the internal engine (8 samples) so this value is further processed as follows:

3.1.2 Harmonic Product Spectrum (HPS) and Cepstrum (Monophonic)

JSyn also includes an implementation of the FFT. Two simple monophonic algorithms have been implemented based

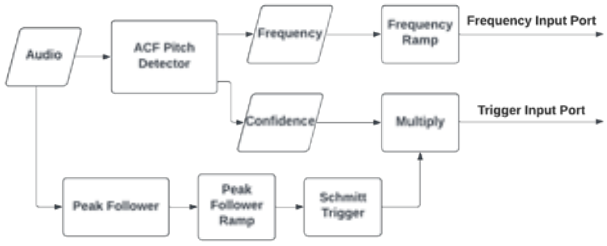


Figure 4. The workflow of the ACF pitch detector.

on the FFT. These methods work best for an input signal that is monophonic and has an harmonic pitched content.

The Harmonic Product Spectrum method was presented by Noll in [7]. The main idea is that the spectrum consists of a series of peaks corresponding to the fundamental frequency and harmonic components positioned at integer multiples of the fundamental. So by downsampling and multiplying the spectrum several times the strongest harmonic peak should emerge. E.g. by downsampling the spectrum by a factor of two, the second peak in the original spectrum will line up with the fundamental.

The use of "cepstrums" was introduce by Noll in [8] in the context of vocal pitch detection. The FFT of the input signal is computed and then its spectrum, in log scale, is treated as a periodic signal itself. The periodicity of this new signal is linked to the spacing between the different peaks. In our implementation we compute the Cepstrum as:

$$C_p = |F\{log(|F\{f(t)\}|^2)\}|^2 \quad (1)$$

The same post-processing used for JSyn ACF is applied here.

Parameters include the chosen method, the FFT Size (default 1024 samples), which in turns determines the Lowest Frequency, and the Highest Frequency (default 1567.98 Hz MIDI Pitch 91 - G6).

3.1.3 FFT-YIN (Monophonic)

The implementation comes from the Tarsos DSP library. The library includes 6 pitch detection algorithms. It is a faster implementation of the YIN algorithm described in [9] and implemented by Matthias Mauch (Queen Mary University, London). The YIN algorithm is a modified version of the autocorrelation method.

A pitch is triggered when it exceeds a confidence threshold.

Parameters include the FFT Size (default 1024 samples) and the Confidence Threshold (default 0.98).

3.1.4 CQT (Mono- and Poly-phonic)

Utilizes the Tarsos DSP implementation of the Constant Q Transform, based on the work of Brown and Puckette [10] and Blankertz [11]. This transform is particularly well suited for musical applications given its frequency-varying resolution that more closely resembles our auditory system. By selecting its starting frequency to match one of the MIDI pitches, and 12 as the number of bins per octave each bin will correspond to a MIDI pitch. This results in

a significant reduction of the number of output bins when compared to FFT.

A notable drawback is the amount of delay introduced. Let us assume that for a Sampling Rate of 44100 we are interested in tracking the pitches starting from E1 (41.21 Hz). If we want to be able to discriminate E1 from F1 (43.65 Hz) a resolution of 2.44 Hz is required. This results in an FFT of at least 18074 samples. The current implementation search for the next power of 2 in order to optimize the computation. So a resulting FFT of length 32768 needs to be computed. Such an FFT introduces a delay of 743 ms, which might not be suitable for real-time audio applications.

An effective solution to mitigate this problem, at the expenses of a larger memory consumption, and setup time, is to compute multiple CQTs with different starting frequencies. By doing so each CQT will have a different delay, decreasing as the frequency increases, as the resolution needed is also decreasing. This means that higher frequencies can be detected faster than lower frequencies. Even for a single note this might mean that the first pitch triggered would be one octave higher than the actual note. In our application this does not cause particular problems as notes are added to the pool, possibly contributing to a new lowest note (sent to the bass channel), or enriching the arpeggio with new lower octaves.

Two other parameters are available for the CQT: Threshold and Spread. Threshold governs how the pre-computed kernels are truncated. A greater sparsity of the kernel matrices results in fewer operations to be computed. Finally, Spread effectively alters the resolution of the transform. A smaller Spread might result in a bigger FFT to be computed. For the previous explanation a value of 1.0 was used.

The current implementation utilizes 1 CQT per octave. Parameters include Lowest Frequency (default 41.205 Hz MIDI Pitch 28 - E1), Highest frequency (default 2637.02 Hz MIDI Pitch 100 - E7), Threshold (default 0.01), Spread (default 0.55, which is the smallest value that does not increase the FFT size compared to a Spread of 1.0).

The lower portion of the GUI depicts in real-time the CQT bins and their amplitude. Bins are colored green when they exceed the threshold (red line) set by the Threshold Slider, resulting in output pitches to be triggered.

A flag parameter called Auto-Tune engages a mode where if up to N contiguous bins (configuration parameter) are active only the one with the highest magnitude triggers an output. This can be useful if the audio input is not in tune with the CQT bins, or if the Spread is increased to reduce the delay resulting in a "blur" between consecutive bins. If more than N consecutive bins are active we consider it a "cluster" and all the pitches are activated.

The magnitude of a bin in relationship with the threshold is used to determine the Velocity of the *noteOn* MIDI message. Two parameters for minimum and maximum velocity are mapped with a linear interpolation to the threshold and a value corresponding to $2 * threshold$. The use of a Velocity based on the strength of the audio input can be further expanded by using the Aftertouch MIDI message (as opposed to a Channel Pressure message as it does not carry the Note information) to alter the volume of a note that is already playing, without triggering the notes in the arpeggio again.

3.2 Arpeggiator Logic

Arpeggiatorum's arpeggiation logic has been designed with flexibility in mind and the possibility to define and play rich arpeggios. Hence, it offers a number of parameters, all editable in the GUI (see Figure 1) and via MIDI control change messages.

A typical arpeggiator takes a number of pitches (MIDI *noteOn* messages, respectively) into a pool and plays them in repeating sequence as long as they are not removed from the pool (via corresponding *noteOff*). So does *Arpeggiatorum*. It collects the input pitches in a list sorted in ascending order, the *note pool*. A real-time thread (using the JSyn scheduler) traverses the note pool in a specified *pattern* (upward, downward, up and down, randomly with and without tone repetitions) and in a specified *tempo*, thus creating the arpeggiation sequence. The note pool is more than a sorted list of notes. The traversal pattern is part of its functionality, thus offering a handy `getNext()` method. Notes can be added to and removed from the pool while the sequence is generated. The traversal functionality ensures that the pattern stays intact.

The *articulation* of the notes can be set from staccato over tenuto up to an overlapping legato play. This is particularly useful when the actions of an organ are rather sluggish and would prevent very fast arpeggios from ever creating a sounding tone. Typical organ keyboards and pedals have a further limitation: They do not offer the full *range* of 128 MIDI pitches. This applies to most other instruments as well. Hence, we added a two-pole range slider to delimit the pitch values in the note pool that are included in the arpeggiation sequence.

So far, arpeggio sequences have only been created from manually triggered notes. A further functionality of the note pool is *tonal enrichment*, i.e. for each manually triggered note it can add up to 15 further notes. These are defined by interval relations with the triggered note. While the GUI of *Arpeggiatorum* allows users to specify their own series of intervals, it is also possible to choose from a list of presets such as "Octaves and Fifths", "Major Triad", "Minor Triad", "Fourths", and "Series of Overtones". A slider also allows to modulate on-the-fly how many of the enrichment notes should be added to the note pool, thus offering a further dimension for creative interaction during performances.

A final element of *Arpeggiatorum*'s logic is aimed directly at the organ with its multiple manuals and pedals. These are typically accessed via different MIDI channels. The arpeggios are sent to one of these. *Arpeggiatorum*, however, also allows sending the triggered notes (i.e. without arpeggiation) to another channel. And the lowest of the triggered notes is automatically transposed down (as bass note) and sent to a third channel, typically the pedals. Each

of the three channels (arpeggio channel, held note channel, and bass channel) can be muted. An organist who has used *Arpeggiatorum* in concerts emphasized this as a particularly useful feature. It allowed him to perform on a short 2-octaves MIDI keyboard in the middle of the nave (WiFi connection to the organ console) but effectively playing on two manuals and pedals.

4. AUDIO INPUT EVALUATION

For testing purposes the application is set to receive audio input from a virtual interface (VB-Cable [1]) connected to a DAW (Reaper [2]). The generated MIDI messages are sent to a software synthesizer (Java's Gervill), and the output is then sent to another virtual interface that is recorded in the DAW. All test input files have the same structure: 3 s of silence as the head and tail of the samples. All samples are normalized at -6dB FS. This allows us to conduct a preliminary evaluation measuring the accuracy of the CQT algorithm. Regarding delay the current setup (with virtual interfaces) introduces 40 msec to the measurements. For each test signal 5 measurements have been recorded.

The use of CQT for pitch-tracking is not in itself new, but in our implementation, aimed at real-time tracking, we use multiple CQTs for different octaves, thus mitigating the delay problem. For this reason we focus our attention to this method. In Table 2 we see the results for a sine wave of 440 Hz. We can observe that in Run 3 the correct bin has

Run	Pitch	Velocity
1	[69] 440Hz	127
	[69] 440Hz	34
2	[69] 440Hz	127
	[69] 440Hz	35
3	[69] 440Hz	121
	[69] 440Hz	127
	[68] 415Hz, [70] 466Hz	41, 39
4	[69] 440Hz	127
	[69] 440Hz	34
5	[69] 440Hz	127
	[69] 440Hz	34

Table 2. 5 Runs of the CQT Algorithm without Auto-Tune on a -6 dB FS sine wave. Spurious pitches have been detected in neighbouring bins.

been detected first with the strongest velocity, followed by the 2 neighbouring bins. For this reason the introduction of the Auto-Tune mode seems a promising approach to mitigate these occurrences. In Table 3 we show the results for the same input, with Auto-Tune activated.

In both examples, the second pitch, with a lower velocity, is triggered at the end of the sine wave.

In Figure 5 we show the view from the DAW of 5 takes for a polyphonic input. We can see that different runs have different delays. Delay depends on the actual frequency content, as a result of running multiple CQTs, and the position within the internal buffer when the input is captured.

Run	Pitch	Velocity
1	[69] 440Hz	123
	[69] 440Hz	30
2	[69] 440Hz	127
	[69] 440Hz	37
3	[69] 440Hz	127
	[69] 440Hz	34
4	[69] 440Hz	127
	[69] 440Hz	34
5	[69] 440Hz	127
	[69] 440Hz	34

Table 3. 5 Runs of the CQT Algorithm with Auto-Tune on a -6 dB FS sine wave.

Figure 5. Example of 5 takes for a test sound, showing different delays.

Synthesized sounds, as well as recordings, mono- and polyphonic, have been employed for testing. Test materials, as well as the settings used for the application, are available on GitHub.

5. CONCLUSIONS AND FUTURE WORKS

With the development of *Arpeggiatorum* we wanted to provide a tool that can be readily employed by organ players, expand playing techniques on organs (ultimately also for other instruments), and foster cooperation and future collaborations. *Arpeggiatorum* is scheduled to be used in concerts and other performances. The valuable feedback provided by musicians will determine how the tool will evolve and what features will be introduced.

The next phase for *Arpeggiatorum* includes both an extension of its core features to address requests received

[1] https://vb-audio.com/Cable/
[2] https://reaper.fm

during the preliminary evaluation, and a more thorough examination of the affordances that are granted by this application. In terms of functionalities we plan to introduce an audio feature extractor that would allow to map input parameters (e.g. roughness) to arpeggiator parameters (e.g. pattern).

The source code for the project is available on GitHub: (https://github.com/axelberndt/Arpeggiatorum).

Acknowledgments

This research is supported by KreativInstitut.OWL [3]: a consortium consisting of OWL University of Applied Sciences and Arts, Detmold University of Music, and Paderborn University, funded by the Ministry of Economic Affairs, Industry, Climate Action and Energy of the State of North Rhine-Westphalia, Germany.

6. REFERENCES

[1] J. W. Robinson and S. L. Howell, "Automatic arpeggiator," *The Journal of the Acoustical Society of America*, vol. 74, no. 5, pp. 1666–1666, 1983. [Online]. Available: https://doi.org/10.1121/1.390092

[2] R. Arar and A. Kapur, "A history of sequencers: Interfaces for organizing pattern-based music," in *Sound and Music Computing Conference, Stockholm, Sweden*, 2013.

[3] M. Lozej, "The Arpeggiator: A Compositional tool for Performance and Production," Master's thesis, York University, Canada, 2016. [Online]. Available: http://hdl.handle.net/10315/32712

[4] P. Burk, "JSyn-a real-time synthesis API for Java," in *Proceedings of the 24th International Computer Music Conference (ICMC)*, 1998.

[5] J. Six, O. Cornelis, and M. Leman, "TarsosDSP, a Real-Time Audio Processing Framework in Java," in *Proceedings of the 53rd AES Conference (AES 53rd)*, 2014. [Online]. Available: http://www.aes.org/e-lib/browse.cfm?elib=17089

[6] R. Bencina and P. Burk, "PortAudio - an open source cross platform audio API," in *Proceedings of the 27th International Computer Music Conference (ICMC)*, 2001.

[7] A. M. Noll, "Pitch determination of human speech by the harmonic product spectrum, the harmonic surn spectrum, and a maximum likelihood estimate," in *Symposium on Computer Processing in Communication, ed.*, vol. 19. University of Broodlyn Press, New York, 1970, pp. 779–797.

[8] ——, "Short-time spectrum and "cepstrum" techniques for vocal-pitch detection," *The Journal of the Acoustical Society of America*, vol. 36, no. 2, pp. 296–302, 1964.

[9] A. De Cheveigné and H. Kawahara, "YIN, a fundamental frequency estimator for speech and music," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1917–1930, 2002. [Online]. Available: https://doi.org/10.1121/1.1458024

[10] J. C. Brown and M. S. Puckette, "An efficient algorithm for the calculation of a constant Q transform," *The Journal of the Acoustical Society of America*, vol. 92, no. 5, pp. 2698–2701, 1992.

[11] B. Blankertz, "The Constant Q Transform," 1999, TU-Berlin, unpublished notes. [Online]. Available: https://www.user.tu-berlin.de/blanker/drafts.html

[3] https://kreativ.institute

SOUNDSCAPE PERSONALISATION AT WORK: DESIGNING AI-ENABLED SOUND TECHNOLOGIES FOR THE WORKPLACE

Thomas DEACON (t.deacon@surrey.ac.uk) [1], **Arshdeep SINGH**[1], **Gabriel BIBBÓ**[1], and **Mark D. PLUMBLEY**[1]

[1] *Center for Vision, Speech and Signal Processing*, **University of Surrey**, UK

ABSTRACT

Poor workplace soundscapes can negatively impact productivity and employee satisfaction. While current regulations and physical acoustic treatments are beneficial, the potential of AI sound systems to enhance worker wellbeing is not fully explored. This paper investigates the use of AI-enabled sound technologies in workplaces, aiming to boost wellbeing and productivity through a *soundscape approach* while addressing user concerns. To evaluate these systems, we used *scenario-based design* and focus groups with knowledge workers from open-plan offices and those working remotely. Participants were presented with initial design concepts for AI sound analysis and control systems. This paper outlines user requirements and recommendations gathered from these focus groups, with a specific emphasis on soundscape personalisation and the creation of relevant datasets.

1. INTRODUCTION

New Artificial Intelligence (AI) technologies offer promising opportunities to improve workplaces by monitoring and controlling environmental sound. Specifically we refer to the AI technology of machine listening that can automatically categorise everyday sounds [1]. Such technology can be used to monitor workplaces to develop new strategies for improving quality of life. Inspired by the *soundscape approach* - where environmental noise researchers and practioners consider the overall acoustic environment including its potential for positive and restorative effects on human health and wellbeing [2] - we research how innovations in machine listening technology could enable the exploration into how context affects people, bringing us closer to understanding how sounds feel, rather than just basing systems on objective measures like loudness. But AI technologies do not come without ethical risks and significant issues around adoption. If such obstacles can be navigated, new soundscape technologies can provide a means for organisations to support wellbeing and improved work environments.

The design area we are addressing in this research is the creation of AI sound sensing and soundscape control systems for the workplace (both offices and at home). We use a parallel design process of deep-learning research [3], co-design [4], and prototyping [5]. So far, we have gathered multiple concepts for the design of AI sound sensing interventions in the workplace. Next, we look to uncover potential adoption issues and further understand workers needs through concept evaluation.

This paper investigates the perception of AI-enabled soundscape technologies for enhancing workers' wellbeing and productivity using scenario-based design [6] and reflexive thematic analysis [7]. We used fictional technology scenarios with both positive and negative viewpoints on the same concepts in structured focus groups. This work borrows from previous design fiction methods used to understand future technologies [8–10]. Our participant recruitment targeted adult knowledge workers, including programmers, researchers, and accountants, across two work contexts: open-plan offices (OPOs) and work-from-home (WFH). The paper's findings present how workers perceive design concepts, highlighting needs and concerns about proposed soundscape systems in the two work contexts. Our study contributes novel findings to Sound and Music Computing (SMC) in the areas of applied audio AI and sonic experience design for work settings.

2. RELATED WORK

2.1 Knowledge Workers, Noise, and Wellbeing

Our work focuses on knowledge/information workers whose type of work often involves "reflective work" such as problem-solving, strategising, and creative development [11]. An issue for reflective work is that it can be influenced by the environment, particularly sound. For example, open plan office (OPO) spaces are often filled with distractions from other people and sounds [12]. Persistent issues with distraction can significantly impact productivity and overall workplace satisfaction [13, 14]. When working from home, sound can also become a distraction [15]. Many home workspaces also lack the common sound-masking capabilities that offices have, which makes workers more susceptible to noise from family, pets, and the outside environment. This can be especially challenging for those who require a quiet space to focus on reflective work, with a common coping strategy being to use headphones to create a sense of privacy [16,17]. Therefore, it is beneficial to create an environment that allows individuals in any environment to concentrate on their tasks without being disrupted by external noise,

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

while also improving the soundscape in order to minimise physiological harms [18].

2.2 AI Soundscape Analysis and Control Systems

AI sound technologies could alleviate the issues of workplace sound to enhance the capacities of workers to engage in reflective work. For instance, evidence suggests that a moderate level of background noise enhances creativity compared to high-, low-, and no-noise conditions [19]. In order to effectively apply this discovery in the workplace, an AI audio service is required to handle several tasks: capturing and analysing sounds on-site, collecting and processing personalisation features, producing contextually relevant audio, and broadcasting the audio within the work environment. Below, we discuss current research related to these areas.

2.2.1 Computational Analysis of Soundscapes

Within computational analysis of soundscapes, Sound Event Detection (SED) and Acoustic Scene Classification (ASC) technologies allow a machine to sense sound in a way comparable to aspects of human hearing, where we perceive sources and objects instead of just signals and mathematical features [20]. Automatic SED is an audio machine learning approach to spot "what is happening in an audio signal and when it is happening" [1], while ASC is the process of labelling an audio recorded in a specific environment e.g. park, large room [21]. Combined, these systems allow contextual understanding of sound, while providing human readable outputs like scene and event classifications. When linked to other supporting systems and technical developments in AI and SMC, such classification technologies provide a basis to explore personalised soundscape systems.

2.2.2 Soundscape Personalisation

A key design aim in our project is to understand the value of individuated soundscapes at work. Individualisation can be attained through a combination of personalisation and customisation [22]. Personalisation tailors content, structure, and presentation to an individual automatically using *adaptivity* [22]. Customisation offers *adaptability* [22], allowing individuals to personally modify content presentation or structure. In this way, personalisation is system-initiated and system-driven, while customisation is user-initiated and user-driven process. An example of an adaptive system is when AI algorithms analyse biometric and behavioural data to create personalised soundscapes that promote focus or relaxation [23]. Already, companies like *Endel* have developed an AI product that generates personalised soundscapes using circadian rhythm and heart-rate to optimise for modes like focus and relaxation. [24]. For Deaf [1] and hard of hearing individuals sonic customisation refers to visualisations based on sound recognition tools built to suit the individual needs and preferences of users, as pre-trained sound recognition models may not meet their diverse requirements [25]. Central to our idea of soundscape personalisation is making systems that can use real-time data and sonic preferences, while understanding user contexts related to specific goals and settings.

2.2.3 Personalised Audio Systems for Work

In the area of workplace noise abatement, personal audio systems exist which can improve speech privacy in shared spaces by focusing speech towards the target listener and masking surrounding sounds, without causing annoyance to nearby individuals [26]. Also, a personalised audio masking signal can significantly enhance speech privacy in office spaces without significantly increasing annoyance when compared to white noise [27]. Previously, a real-time natural soundscape generation system was proposed for workspace voice-masking [28]. This system, based on current weather conditions, can improve speech privacy and reduce distractions, while maintaining a pleasing and informative environment. *Sound Bubbles* are a way of creating a personalised auditory environment by generation and modification of sound sequences around a worker, that are in line with the physical environment, enhancing their ability to concentrate [29–31].

Within audio engineering, extensive research has been conducted into *Sound Zones* as a way to create physically limited areas within a room where sounds can be controlled, allowing for multiple personal listening experiences within the same space without the need for headphones or disturbing others [32–35]. Since that foundational work, Sound Zones have been developed in domestic audio interaction design settings [36–41]. Sound Zones have recently been adapted to hospitals where the value of individualised sound environments in shared spaces is proposed and evaluated [42].

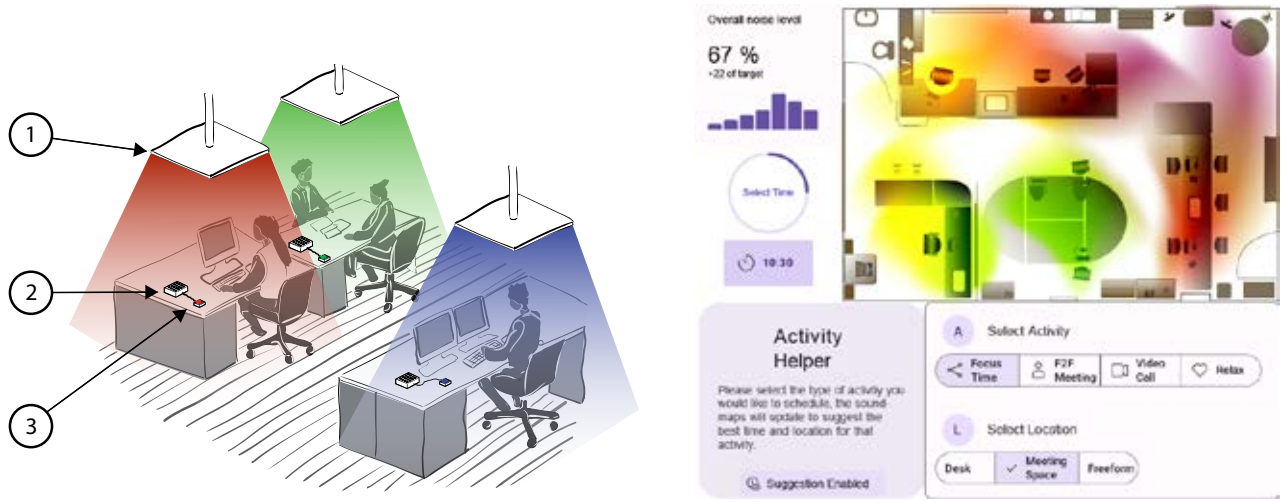
3. DESIGN CONCEPTS FOR WORKPLACE AI SOUND TECHNOLOGIES

Based on requirements developed in our previous co-design work [4], our workplace AI sound technology design concepts are based on technological possibility within the state-of the art in machine listening [43], sound engineering [40], workplace soundscape design [44], and workplace wellbeing AI [45]. The following design proposals include the technologies located in soundscape understanding, personalised sound control, and novel audio reproduction. These are imaginary conceptual prototypes: we did not build physical systems for this study. Conceptual prototypes provide a quick method of iterating a design space focused on human-centred design.

3.1 Sound Analysis Concepts

Sound Analyser: A sound analyser is combination of microphones and a sound labelling device, that feeds an AI model, that once trained can estimate whether sounds are pleasant and annoying in the workplace, as well as report sound levels. The purpose of Sound Analyser is to gather information about the stress and pleasure caused by

[1] We use Deaf with a capital D to refer to people who have been deaf all their lives, or since before they started to learn to talk. See https://www.diversitystyleguide.com/glossary/deaf-deaf/



(a) Concept prototype of AI sound system that creates personalised Sound Zones that integrate preferences. Element numbered are 1: Sound Zones via Highly Directional Speakers, 2: Sound Analyser Microphone Array, 3: Sound Analyser Preference Input.

(b) Concept prototype for AI Sound Map system that indicates suitable places to do certain types of work at different times of the day. Colours indicate suitability for a given task, green-yellow indicate good, red-purple bad.

Figure 1. Concept prototypes

sounds. It can provide data for decision making and help a variety of stakeholders understand types of sounds in a space. A simple rendering of a sound analyser box and preference input system can be found in Fig. 1a. A key user action of this system is to annotate sounds in some way: each scenario offers different means of doing this.

Sound Mapping of Spaces: Sound Maps are a visual way to represent noise levels and sound wellbeing impacts within certain contexts. For example, highlighting whether a given time and place are suitable for focused deep work given the user's preference profile. This idea is represented in Fig. 1b, which is a mock-up of a UI (the system is fictional). Preference profiles would include user specific recommendations based on computation of expected sound levels and timbre profile (e.g. a lot of speech being distracting). A further use of a Sound Map is to measure office space appropriateness and support improved sound etiquette in workplaces.

3.2 Sound Control Concepts

Adaptive Sound Masking and Artificial Soundscapes: Adaptive Sound Masking involves generating noise to "mask" other sounds, using personalised profiles and situational audio analysis. Its purpose is to improve productivity by reducing distractions in certain areas, protecting speech privacy, and generally improving office soundscapes. Artificial Soundscapes involve relaying or generating sounds from other locations, such as a beach or a rainforest. The purpose of Artificial Soundscapes is to improve productivity/wellbeing by creating a positive and calming ambience, signalling time of day, and reducing distraction from noise.

Personalised Audio via Highly Directional Sound (HDS): This concept involves the use of highly directional speakers, mounted on the ceiling or desk, to create Sound Zones such as those pictured in Fig. 1a. Sound Zones are designed to provide sound that can only be heard by one person in mixed-use spaces [42]. Primarily, they serve as reproduction technology for Adaptive Sound Masking and Artificial Soundscapes.

4. METHODOLOGY

4.1 Study Aim

The aim of our study was to gain worker insights on audio AI in open-plan office (OPO) and work-from-home (WFH) settings, to understand needs for AI-driven soundscape wellbeing and productivity systems.

4.2 Exploring Future Technologies with Stories

We use scenario-based design (SBD) to avoid fixating on solutions and ensure a better fit with the problem space. SBD engages knowledge workers in the design process [6]. While SBD has been applied to explore home autonomous systems [9], there is a lack of evaluation using SBD for AI audio technology in smart buildings, making our research a novel contribution to applied sound sensing AI. Our SBD evaluation involves stories, illustrations, and a structured feedback workshop. Inspired by the *contravision* approach [8], we present positive and negative stories to uncover concerns related to the design and acceptance of futuristic personal technologies. This approach is well-suited for understanding sensitive issues in pervasive system design [8]. Similar provocative design methods have recently been used for soundscape research in the SMC field [10].

In *contravision* scenarios, participants' preference for a specific rendering is not crucial. Instead, we focus on gathering information about broader concepts that influence engagement and understanding of technology [8]. *Contravision* scenarios provoke discussion and allow participants to reflect on their workplace expectations [9]. The ability to articulate the implications, provide

comments on acceptability, and envision the consequences of such systems, is more important than a preference for a particular technology [46]. This facilitates discussions on values, fears, desires, and understanding, of technology; providing insights for design proposals, research challenges, use-cases, and deployment concerns.

4.3 Scenarios and Narrative Context

The scenarios are stories about AI sound systems in a spatial setting (OPO/WFH). Each scenario set (OPO/WFH) has a different protagonist who represents a typical knowledge worker. Their identity is established before the scenario presentations. The protagonist's preferences and environmental context are provided to help the reader engage with the story setting.

Both the OPO and WFH scenarios resembled each other in terms of core content. This is done by setting out the narrative "claims" that the stories reflect [6, 8]. Table 1 highlights the structural elements of the scenario sets. However, scenarios do not have to have the same ordering of these elements. The wording of the scenarios also varied based on work context (OPO/WFH). Scenario set A explores Sound Analysis concepts, while Scenario set B covers Sound Control concepts. All scenarios are available online [2].

4.4 Participants Screening and Selection

Twenty-two knowledge worker participants were recruited using University mailing lists, social media, and word of mouth. Participants for focus groups have two groupings: OPO (N=10); WFH (N=12). The OPO group was an in-person event while the WFH took place remotely on MS Teams. General screening criteria included being an adult (over 18) and their job, or work experience, involves problem-solving, research, or creative ideation (to qualify as knowledge worker). The OPO group were open-plan office workers within 15 miles of University of Surrey campus. The WFH group were remote workers residing within 3 hours of the UK time zone. Participants were selected via purposive sampling based on job type, age breadth, and work context. Demographics of each group were: *OPO* - Sex: Male (N=3), Female (N=7); Age: 25-39 (N=5), 40-60 (N=4), 60+ (N=1). *WFH* - Sex: Male (N=5), Female (N=7); Age: 18-24 (N=3), 25-39 (N=7), 40-60 (N=2). All participants self-reported having functional hearing, ranging from poor to excellent. All participants provided informed consent and received a £40GBP voucher for their time. The study received favourable ethical approval, ID: FEPS 22-23 015 EGA Amend 1.

4.5 Focus Group Process

Focus groups lasted for two hours. First we introduced the concepts of AI sound technology and set up the story context. We then used a structured feedback process for each scenario set (A & B). This process included 2 phases of *Scenario Read and Reflect* each followed by a *Group Discussion*, then concluded with an *Open Discussion*. In the following sub-sections, we describe these phases in detail.

Scenario Read and Reflect: Participants were given scenarios to read individually in silence. After reading, participants reflected individually on the scenarios using the following questions:

1. What are your thoughts on working in an environment like this?
2. What are your thoughts on the believability of this scenario?
3. How would you use the system in this scenario?
4. Scenario Set Specific Question:
    - **Set A Only**: What do you think about the Sound Map in this scenario?
    - **Set B Only**: What are your thoughts on the sound systems in this scenario?

Group Discussions: These were conducted with smaller sub-groups of 3-5 people to encourage active participation and gather diverse information. Each sub-group scribe (a research team member) reported back to the whole group, and facilitated group discussion.

Open Discussions: Before the open discussion, key feedback from each sub-group is introduced by group scribes. After that, the open discussion was allowed to evolve in an open-ended manner supported by the facilitator.

4.6 Data Collection

Data collection for the focus groups included audio and video recordings, transcriptions, field-notes, and booklet responses. In-person groups used physical booklets for scenarios and question answers, and discussions were video recorded. Remote groups used an online form for scenarios and question answers, and discussions were recorded in MS Teams rooms. Booklets, notes, and memos were scanned and summarised in Word files. All the data was consolidated for analysis in MAXQDA24, a software program used for qualitative data analysis [47].

4.7 Analysis Methods

The data from booklets and discussions was analysed using reflexive thematic analysis (RTA) [7] to identify the reasoning and topics put forward by the participants. RTA is a method for identifying and analysing themes in qualitative data related to various research questions. RTA can investigate individual users' concerns while also addressing themes throughout the research process. Following best practice [7], we do not report numbered counts when reporting themes. A theme is considered important in relation to an analysis question, and it may only occur once to be significant. Instead, the relevance of the finding to the research focus is emphasised. For this paper, we report only themes related to the design concepts proposed in Sec. 3 and the design goal of soundscape personalisation introduced in Sec. 2. This choice is based

[2] https://adjuvant.github.io/SAW-FG-SMC24-Stories/

Table 1. Scenario Claims for Design Concepts

Scenario Element	*Positive Story Claims (Utopia)*	*Negative Story Claims (Dystopia)*
Set A Claims: Sound Analysis Concepts Scenario Elements. Sound Analyser (SA), Sound Map (SM).		
SA Devices	Handheld device	Desktop box, web app with notifications
SM Devices	Installed microphone array, room visualisation	Installed microphone array, room visualiser, LED alarms
SA Purpose	Build SM preference profile of what is pleasant or annoying	
SM Purpose	Informing sound etiquette policy	
Receiving the SA	Human Resources wellbeing initiative, everyone does tagging	HR wellbeing initiative, specific people given a tagging task
Using the SA	Easy, just point at something and label it (control, agency), engages you in moving around the office	Forced to figure out placement and system settings. Real-time tagging, system requires you to go through sounds and label them seemingly at random (human-in-the-loop).
Experience of tagging sounds	Strange at first but satisfying, new appreciation for sounds is mindful	Notice mostly annoying sounds, feels strange to label positive ones in the moment.
Impact of SM	Shows good and bad places/times for work activities. Improved office vibe	Feeling of being judged/judging others
Set B Claims: Sound Control Concepts Scenario Elements		
How the system is installed	Personalised device, engineer installation, software calibration, control app, desk-based	Workspace-wide array of overhead speakers
System description	Requires training via annotations, subscription service. System can relay sounds.	System installed without prior knowledge/consent, capturing multi-modal data seemingly without permission
Soundscape change	Gentle augmentation of soundscape. Benefits to focus.	Unforeseen adverse effects of the AI sound sensing technology, the system experiments on you to determine preferences (fully adaptive)
Reflection	Satisfaction and no reality dissonance	Discomfort, anger, social confrontation, assertion that technology can go too far
Novel elements	Doubt at the system initially, system has a set of moods that can be extended, social brainstorming presents a problem for different preferences	Chatbot service, profile config, tech support need to amend the system when consent status changed.

on the page limit and relevance to SMC community. Broader thematic analysis findings will be reported in future publications.

5. THEMATIC ANALYSIS: WORKPLACE SOUNDSCAPE PERSONALISATION

The following research questions guided the RTA process:

1. How do participants understand the purpose of current prototypes/design concepts?
2. What concerns do participants have about such technology in the OPO workplace or WFH?
3. How are AI personalisation and customisation technologies received?

In the following analysis, themes are denoted using sans-serif font, italics and bold case, and capitalised words e.g. ***Theme Title***, and we provide references to the sub-themes on which top-level themes are based. When a theme is related to a specific scenario set (A: Sound Annotator; B: Sound Controller) it is indicated in square brackets in the theme title e.g. [A] or [B]. For the presentation of any data, rather than use verbatim, we edited many extracts to improve readability, however we did not alter keyword choice or intention.

5.1 Open-plan Office (OPO)

Personalisation Issues Participants highlighted issues with personalisation in OPO settings. The theme involves concerns about the reliability of sound tagging, level of detail in personalisation, delivery of personalised sounds via networked speaker systems in shared spaces. We highlight some of the contributing sub-themes below.

Scepticism Of Personalisation Claims It was suggested that people have different preferences and perceptions of sounds which felt incongruous with the claims

around personalisation: *"Different people perceive sounds differently"* (P2). Also, there was the belief that systems would not be granular enough to handle conflicting noise preferences in office spaces: *"in a communal office there would be too much contradicting noise based of numerous peoples preferences."* (P4). Scenarios around sound control were not accepted by some as possible as it was not understood *"how different people's soundscapes, played through speakers could co-exist in the same environment"* (P1). Overall, the benefits of systems felt unclear and many participants did not find proposals believable.

Limited Usefulness And Appeal Of AI Participants suggest that some AI sound systems, such as Sound Maps and personalised speaker systems, may not be worth using due to triviality of applications, lack of applicability in certain spaces like meeting rooms or shared public areas, and preferences for personal control through headphones. For example: *"I think it is useful if there is enough spaces; only if you actually have many zones with different purposes for them, because I feel like in most offices, every human can create a sound map in their head: open space is noisy, meeting rooms for private conversations and video calls"* (P1). Some participants propose simpler solutions, such as physical signage for indicating quiet areas, rather than relying solely on AI sound analysis and visualisations *"to tell people the expectations of certain areas"* (P4). Personalised HDS systems were considered too expensive for most workplaces. Also it was suggested that systems *"will work on headphone instead of speaker"* (P10), highlighting simpler solutions for public areas not using networks of speakers that also provide more control, *"I can use my own headphones to listen to sounds I like when I want, rather than being forced to hear something all the time"* (P6). Overall, participants express concerns about the limited usefulness and appeal of AI sound systems for controlling and analysing sounds in work situations, and propose simpler solutions as more effective alternatives.

5.2 Work-from-home (WFH)

Productivity Benefits This theme explores the potential benefits of AI audio systems in the home workplace. Participants express their belief in the adoption potential of these systems, highlighting positive impacts on personal productivity and the creation of a comfortable work environment. For example, P18 responds to Sound Control systems questions: *"In an environment where I need more of attention and focus I would consider it"*. Participants believed sound analysis reduces distractions and increases efficiency by optimising task timing and location, potentially enhancing comfort and productivity. Some participants believe sound mapping could optimize home office environments. For example, P17 stated their preference for *"staying in different environments based on which sounds are most conducive and productive"*.

In terms of Sound Control Systems, participants recognise the positive impact of a well-designed system on comfort and wellbeing. However, they express concerns about disruption and distraction caused by poorly designed systems. Reliability and accuracy are identified as essential features, ensuring consistent and effective sound control. Participants also emphasise the need to optimise sound environments for both productivity and overall wellbeing.

Desire For Personalisation And Customisation This theme highlights behaviours and needs around personalisation. Participants find personalised sound environments innovative and helpful, with one participant noting that *"it's quite conducive and relaxing since I have control of the sound around me"* (P16). The personalised aspect is appealing to participants, who think they may feel less stressed when using AI sound tools: *"I like that this environment is centred around the person. It feels like an ambient and relaxing environment"* (P17). Participants illustrate a desire for personalisation to create a good work environment to *"feel comfortable and welcomed"* (P13) as systems *"allow me to control all forms of noise"* (P13).

The *"seamless nature"* (P17) of personalised sound control is seen as an evolution of existing technologies, such as YouTube and lo-fi music, and allows individuals to have more control over their work environment by having *"sounds rise and fall at different times"* (P17). Participants appreciate the ability to experiment with different sounds and customise their auditory environment to meet their individual preferences and needs: *"The sounds could be adjusted to reflect my behaviour and work style"* (P22). Participants voiced a preference for systems that could be easily personalised to their individual needs, and where they could maintain a level of control over operations (customisation), with one participant noting that *"the mood enhancement is really the best thing here"* (P12).

Some worry about stress and dissatisfaction if the system doesn't cater to their specific preferences, such as blending preferences of other household members' likes and dislikes. Furthermore, participants express a desire for personalised sound settings, including saving and labelling preferred sounds, setting up individual preferences, analysing sounds around them, and adjusting background noises to specifications or moods. The ability to input and incorporate one's own sounds into the system is seen as appealing, potentially increasing self-awareness of sounds and personal behaviour. This theme highlights the role of individual user preference control in influencing acceptance and emotional responses to the system.

5.3 Shared Perspectives

Annotation Problematic Both groups found manual processes for labelling to be strange choices, for a variety of reasons. Part of this was finding the suggested methods for tagging sounds as annoying, while another aspect was that some did not think methods of annotation would be that useful and might create bad data. OPO participants question the accuracy of datasets and practicality proposed analysis devices to tag sounds. This was due to concerns about forgetting to use sound analysers/taggers and the subjectivity of data and potential biases. Some participants express scepticism that OPO devices and workers can accurately capture sound in a work environment due to the fleeting nature of sounds and the potential for individuals to focus on their tasks rather than tagging sound events.

Another concern was impact on neurodivergent individuals if datasets are too biased towards normative sensory experiences. Example contributions: *"For someone like me, I might forget about the sound analyser when I'm focused on work. Therefore, it doesn't provide an accurate sound analysis of my space. In a week, regardless of how many times I use it, I sometimes forget to take it with me."* (P11), *"I'm not sure anyone would enjoy tagging things, there is a novelty value at first but this would soon wear off."* (P9), and *"I would be concerned that people would purposely listen out for sounds to tag but whether these sounds would be distracting for their work is another matter."* (P9). Overall, participants express concerns about the practicality and effectiveness of manual sound event annotation to make data of sufficient quality.

Distraction & Annoyance This theme is closely linked with the previous ***Annotation Problematic*** theme but includes aspects of all scenarios. Unsurprisingly, most participants preferred utopian scenarios for their less intrusive nature, highlighting the distracting and unpleasant nature of systems in sound control dystopia scenarios. For both analysis and control technologies, there were concerns about the practicality of implementing such systems in communal workspaces and the potential for conflicting noise preferences, which was perceived as annoying. Disruption caused by notifications, flashing box/lights, and need to tag on demand were a key distractions. Related to the previous theme on annotation, it was suggested that tagging was a annoying task interruption: *"The sound analyser was quite disruptive to my workflow because I constantly had to hold the device to tag and label sounds instead of focusing on my work."* (P11). Overall, the participants felt that the systems could be helpful but the methods of customisation offered and the ways the systems behaved could be frustrating.

Spatial Challenges This theme addresses the obstacles of implementing AI audio in various workplace environments, including shared spaces, the usability of spatial policy systems, and personal space in WFH.

OPO Spatial Policy Usability This theme highlights the practicality of implementing sound maps in different working environments where the function is to inform new policy or adjust behaviour. Concerns are raised about the size and layout of a workspaces (e.g. *"it makes sense only if the space actually offers high variety of zones with different purposes"* (P1)), how space is associated teams and activities, and the system only really applies to workplaces that use hot-desking. Combined, the theme questions the benefits derived from integrating the systems, as it requires everyone to work together and may not leave room for ad-hoc activities. An implication of this is that systems could create cliques and build intolerance.

OPO Impact Of Sound Control On Shared Spaces For OPO, customising soundscapes in shared spaces introduces both challenges and potential conflicts. While some participants see the potential for personalised soundscapes, *"maybe it could work in some public space but not in a packed office"* (P10), it is suggested that conflicts could arise due to varying preferences and spatial setting, *"we have allocated work space and we are sitting right next to each other would blend preferences between users"* (P2). Also it is inferred that accommodating diverse preferences in shared spaces will be a challenge. Some suggest seeking input from aurally diverse groups *"how does it impact on hard of hearing person?"* (P5), and that sound control systems have potential for over-stimulation of neuro-divergent individuals. The generally applicability of systems is questioned, given the perceived difficulty of accommodating varying preferences *"can be difficult in public spaces"* (P7). Additionally, there is concern about creating more noise and annoyance in shared spaces.

WFH Living With AI The use of AI in a WFH setup raises concerns about privacy and work-life balance. While people are comfortable with personal assistants like Alexa, a work-only system may be a barrier to adoption. The potential pitfalls include collecting extraneous data from others and invading personal space. Some suggest restricting the system to only function in the work area and register only data from the worker and not other domestic residents. Overall, the system feels invasive and raises privacy concerns so simple configuration controls are needed for any such analysis or control systems.

6. IMPLICATIONS FOR DESIGN

Many participants see benefits in workplace AI audio systems, yet also identified challenges that need addressing in further research. OPO participants were quite sceptical of proposed systems in utopian and dystopian scenarios, highlighting the need to consider the social implications of AI audio systems. WFH participants showed a strong interest in personalising their auditory environment for mood enhancement, comfort, and the elimination of unwanted sounds. There were noted challenges in implementing systems in shared space, particularly due to conflicting preferences. This suggests a need for solutions that can manage shared spaces' complexities while still offering individual personalisation options.

User preferences and customisation were highlighted as important factors in AI audio system design. The preference for traditional sound management methods, like headphones and personal control over the auditory experience, indicates a potential direction for integrating personalisation into existing practices. Yet, it also emphasises the need to revisit the experience design of personalisation in the workplace more generally. Hence, we recommend that research on AI soundscape personalisation in workplace should focus on diverse data collection, individual control, and navigate the unique challenges of shared environments.

6.1 How can we iteratively train AI and not frustrate users?

The use of tagging and inputting preference data in scenario systems may be distracting and intrusive, leading users to believe that the proposed systems are too much work. Additionally, participants believe that they will use analysis and annotation functionalities in unreliable ways,

which can skew the datasets used to train AI systems. This calls for the research community to develop methods for calibrating datasets for specific use cases and to provide convenient ways for users to input their own data and adjust preferences. When it comes to audio AI, we must address the issue that sound is a fleeting medium, and work technologies should not distract people from their core purpose while at work. Also as AI systems require large amounts of data, we must think about methods to scale data collection. We offer two opportunities below.

6.1.1 Gathering Detailed Information on Workers' Indoor Soundscape Needs

Datasets need to accommodate different people's sound preferences and aural diversity. Here are two suggestions:

Explore more methods of soundscape descriptor elicitation. There is a lack of research into soundscape descriptors for specific task contexts. Additionally, existing indices for indoor soundscapes may not be comprehensive enough for application in workplace settings.

Engage in participatory design for future systems. By setting specific goals and contexts, such as a WFH productivity system or a public space wellbeing soundscape generator, requirements can be determined through co-design and elicitation methods.

6.1.2 Use Prediction Models

Given the opinion of many participants was that in-the-moment tagging would be unreliable for behavioural reasons, an alternative design solution for is to leverage recent research on AI prediction of soundscape annoyance [48]. Here a baseline annoyance model could be tuned in context utilising continuous learning methods [49]. Currently the problem with this is that the annoyance prediction model is designed for outdoor soundscape research, not task-oriented indoor settings. Also a prediction model approach does not furnish us with ways of understanding work context to specify what is annoying when, but it may provide more immediate usability for a given set of general work modes e.g. a focus mode [50].

6.2 *Contraindications* for Design of Sound Control Systems in the Workplace

As highlighted in the ***Distraction & Annoyance*** theme, the claims of systems presented in dystopian sound control scenarios were rejected by most participants in both groups. This provides us with some clear counter-productive directions for future design, namely:

- Do not built expensive infrastructure e.g. networked HDS speaker arrays, when more behaviourally compatible and cost-effective solutions may suffice.
- Make sure wellbeing and productivity interventions are discussed with workers before deployment. We emphasise participatory design to bridge this gap.
- Think about what data is required to make a system function (i.e. is face tracking needed), and ask, can we use less?
- Think about the level of interactivity that is really needed for soundscape augmentation systems: do we really want fully automated, personalised systems constantly collecting data?
- Make sure designed systems have appropriate support for user understanding, control, and resolution of complaints/technical issues.

We hope these contraindications show the need for careful consideration when designing AI systems for peoples' lives, and also demonstrate that design fiction can offer insights into potential pitfalls that need attention.

7. CAVEATS AND LIMITATIONS

Our scenarios were designed to be provocative and elicit extreme value reflections from participants. While this form of human-centered design is still a valid exercise, it is important to balance it in the design space where we aim to improve people's work lives. To do this, follow-up exercises will provide more detail on the forms of desired support in work environments, with or without AI.

Our work only addressed OPO and WFH as separate environments. We did not consider how to frame hybrid work within the context of our scenarios. Additionally, we only conducted two English-language focus groups in a Global North, University-educated setting. As such, the positionality of our work should not be considered universal.

8. CONCLUSION

This paper presents an exploration of AI-enabled sound technologies in the workplace, focusing on how the field of SMC can enhance worker wellbeing and productivity through personalised soundscapes. Through focus groups involving open-plan office (OPO) and work-from-home (WFH) workers, the study uncovers insights into user requirements, challenges, and the potential of AI to create conducive or distracting work environments. Key findings indicate a strong desire for WFH participants for personalised sound environments to improve mood, comfort, and focus by mitigating unwanted noise. However, challenges such as the feasibility and cost of implementing highly directional sound systems in OPO settings, and the need for nuanced data collection and predictive models for effective personalisation, are highlighted. The paper promotes user control and customisation in sound control systems, considering the diverse needs across different work settings. By addressing the limitations and offering design recommendations, this research opens up new possibilities for enhancing workplace wellbeing and productivity through AI-enabled soundscapes.

Acknowledgments

This research was supported by Engineering and Physical Sciences Research Council (EPSRC) Grant EP/T019751/1 "AI for Sound".

9. REFERENCES

[1] A. Mesaros, T. Heittola, T. Virtanen, and M. D. Plumbley, "Sound Event Detection: A tutorial," *IEEE Signal Processing Magazine*, vol. 38, no. 5, pp. 67–83, Sep. 2021.

[2] Ö. Axelsson, M. E. Nilsson, and B. Berglund, "A principal components model of soundscape perception," *The Journal of the Acoustical Society of America*, vol. 128, no. 5, pp. 2836–2846, Nov. 2010.

[3] A. Singh and M. D. Plumbley, "A Passive Similarity based CNN Filter Pruning for Efficient Acoustic Scene Classification," in *Interspeech 2022*, 2022.

[4] E. Corrigan-Kavanagh, A. Fernandez, and M. Plumbley, "Envisioning Sound Sensing Technology for Enhancing Urban Living," in *Environments By Design*. Architecture Media Politics Society, Jan. 2022, p. 11.

[5] G. Bibbó, A. Singh, and M. D. Plumbley, "Recognise and Notify Sound Events using a Raspberry PI based Standalone Device [Demo]," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA 2023)*, New Paltz, NY, USA, Oct. 2023.

[6] M. B. Rosson and J. M. Carroll, "Scenario-Based Design," in *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, J. Jack and A. Sears, Eds. Lawrence Erlbaum Associates, 2002, pp. 1032–1050.

[7] V. Braun and V. Clarke, "Reflecting on reflexive thematic analysis," *Qualitative Research in Sport, Exercise and Health*, vol. 11, no. 4, pp. 589–597, Aug. 2019.

[8] C. Mancini, Y. Rogers, A. K. Bandara, T. Coe, L. Jedrzejczyk, A. N. Joinson, B. A. Price, K. Thomas, and B. Nuseibeh, "Contravision: Exploring users' reactions to futuristic technology," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '10. New York, NY, USA: Association for Computing Machinery, Apr. 2010, pp. 153–162.

[9] T. Nilsson, A. Crabtree, J. Fischer, and B. Koleva, "Breaching the future: Understanding human challenges of autonomous systems for the home," *Personal and Ubiquitous Computing*, vol. 23, no. 2, pp. 287–307, Apr. 2019.

[10] R. Atienza, H. Lindetorp, and K. Falkenberg, "Playing the Design: Creating Soundscapes through Playful Interaction," in *20th Sound and Music Computing Conference (SMC 2023)*. Stockholm, Sweden: Sound and Music Computing Network, Jun. 2023.

[11] W. Reinhardt, B. Schmidt, P. Sloep, and H. Drachsler, "Knowledge Worker Roles and Actions—Results of Two Empirical Studies," *Knowledge and Process Management*, vol. 18, no. 3, pp. 150–174, 2011.

[12] A. Haapakangas, V. Hongisto, J. Hyönä, J. Kokko, and J. Keränen, "Effects of unattended speech on performance and subjective distraction: The role of acoustic design in open-plan offices," *Applied Acoustics*, vol. 86, pp. 1–16, 2014.

[13] H. Jahncke, S. Hygge, N. Halin, A. M. Green, and K. Dimberg, "Open-plan office noise: Cognitive performance and restoration," *Journal of Environmental Psychology*, vol. 31, no. 4, pp. 373–382, Dec. 2011.

[14] Y. Al Horr, M. Arif, A. Kaushik, A. Mazroei, M. Katafygiotou, and E. Elsarrag, "Occupant productivity and office indoor environment quality: A review of the literature," *Building and Environment*, vol. 105, pp. 369–389, Aug. 2016.

[15] G. E. Puglisi, S. Di Blasio, L. Shtrepi, and A. Astolfi, "Remote Working in the COVID-19 Pandemic: Results From a Questionnaire on the Perceived Noise Annoyance," *Frontiers in Built Environment*, vol. 7, p. 688484, 2021.

[16] S. Torresin, R. Albatici, F. Aletta, F. Babich, T. Oberman, A. E. Stawinoga, and J. Kang, "Indoor soundscapes at home during the COVID-19 lockdown in London – Part I: Associations between the perception of the acoustic environment, occupantś activity and well-being," *Applied Acoustics*, vol. 183, p. 108305, Dec. 2021.

[17] S. Torresin, E. Ratcliffe, F. Aletta, R. Albatici, F. Babich, T. Oberman, and J. Kang, "The actual and ideal indoor soundscape for work, relaxation, physical and sexual activity at home: A case study during the COVID-19 lockdown in London," *Frontiers in Psychology*, vol. 13, p. 1038303, Dec. 2022.

[18] T. Münzel, T. Gori, W. Babisch, and M. Basner, "Cardiovascular effects of environmental noise exposure," *European Heart Journal*, vol. 35, no. 13, pp. 829–836, Apr. 2014.

[19] R. Mehta, R. Zhu, and A. Cheema, "Is noise always bad? Exploring the effects of ambient noise on creative cognition," *Journal of Consumer Research*, vol. 39, no. 4, pp. 784–799, 2012.

[20] T. Virtanen, M. D. Plumbley, and D. Ellis, *Computational Analysis of Sound Scenes and Events*. Berlin, Germany: Springer International Publishing, 2018.

[21] B. Ding, T. Zhang, C. Wang, G. Liu, J. Liang, R. Hu, Y. Wu, and D. Guo, "Acoustic scene classification: A comprehensive survey," *Expert Systems with Applications*, vol. 238, p. 121902, Mar. 2024.

[22] H. Treiblmaier, M. Madlberger, N. Knotzer, and I. Pollach, "Evaluating personalization and customization from an ethical point of view: An

empirical study," in *37th Annual Hawaii International Conference on System Sciences, 2004. Proceedings of The*, Jan. 2004.

[23] G. Loudon, D. Zampelis, and G. Deininger, "Using Real-time Biofeedback of Heart Rate Variability Measures to Track and Help Improve Levels of Attention and Relaxation," in *Proceedings of the 2017 ACM SIGCHI Conference on Creativity and Cognition*. Singapore Singapore: ACM, Jun. 2017, pp. 348–355.

[24] T. Hayes, "The science behind Endel's AI-powered soundscapes," https://www.amazon.science/latest-news/the-science-behind-endels-ai-powered-soundscapes, Nov. 2020.

[25] S. M. Goodman, P. Liu, D. Jain, E. J. McDonnell, J. E. Froehlich, and L. Findlater, "Toward User-Driven Sound Recognizer Personalization with People Who Are d/Deaf or Hard of Hearing," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 5, no. 2, Jun. 2021.

[26] D. Wallace and J. Cheer, "Design and evaluation of personal audio systems based on speech privacy constraints," *The Journal of the Acoustical Society of America*, vol. 147, no. 4, pp. 2271–2282, Apr. 2020.

[27] A. Krasnov, E. R. Green, B. Engels, and B. Corden, "Enhanced speech privacy in office spaces," *Building Acoustics*, vol. 26, no. 1, pp. 57–66, Mar. 2019.

[28] C. Riggs and J. Braasch, "Real-time natural soundscape generation based on current weather conditions for workspace voice-masking," *The Journal of the Acoustical Society of America*, vol. 140, no. 4_Supplement, p. 3041, Oct. 2016.

[29] M. L. Eriksson and L. Pareto, "Sound Bubbles for Productive Office Work," in *Nordic Contributions in IS Research*, ser. Lecture Notes in Business Information Processing, U. Lundh Snis, Ed. Cham: Springer International Publishing, 2016, pp. 29–42.

[30] M. L. Eriksson, L. Pareto, R. Atienza, and K. F. Hansen, "My Sound Space: An attentional shield for immersive redirection," in *Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion*, ser. AM '18. New York, NY, USA: Association for Computing Machinery, Sep. 2018.

[31] M. Ljungdahl Eriksson, L. Pareto, and R. Atienza, "The Sound Bubble : An Aesthetic Additive Design Approach to Actively Enhance Acoustic Office Environments," in *13th Conference on Sound and Music Computing*. Zentrum für Mikrotonale Musik und Multimediale Komposition (ZM4), Hochschule für Musik und Theater, 2019, pp. 253–260.

[32] J. Francombe, P. Coleman, M. Olik, K. Baykaner, P. J. B. Jackson, R. Mason, M. Dewhirst, S. Bech, and J. A. Pederson, "Perceptually Optimized Loudspeaker Selection for the Creation of Personal Sound Zones," in *52nd International Conference: Sound Field Control - Engineering and Perception*. Audio Engineering Society, Sep. 2013.

[33] P. Coleman, P. Jackson, M. Olik, and J. A. Pedersen, "Optimizing the Planarity of Sound Zones," in *52nd International Conference: Sound Field Control - Engineering and Perception*. Audio Engineering Society, Sep. 2013.

[34] M. Olik, J. Francombe, P. Coleman, P. J. B. Jackson, M. Olsen, M. Møller, R. Mason, and S. Bech, "A Comparative Performance Study of Sound Zoning Methods in a Reflective Environment," in *52nd International Conference: Sound Field Control - Engineering and Perception*. Audio Engineering Society, Sep. 2013.

[35] P. Coleman and P. Jackson, "Planarity-Based Sound Field Optimization for Multi-Listener Spatial Audio," in *2016 AES International Conference on Sound Field Control*. Audio Engineering Society, Jul. 2016.

[36] S. S. Johansen and P. A. Nielsen, "Personalised Soundscapes in Homes," in *Proceedings of the 2019 on Designing Interactive Systems Conference*, ser. DIS '19. New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 813–822.

[37] S. S. Johansen, P. A. Nielsen, and J. Kjeldskov, "Interaction Design for Domestic Sound Zones," in *Proceedings of the 14th International Audio Mostly Conference: A Journey in Sound*, ser. AM'19. New York, NY, USA: Association for Computing Machinery, Sep. 2019, pp. 248–251.

[38] S. S. Lundgaard, P. A. Nielsen, and J. Kjeldskov, "Designing for domestic sound zone interaction," *Personal and Ubiquitous Computing*, vol. 26, no. 4, pp. 1225–1236, Mar. 2020.

[39] S. S. Johansen, P. A. Nielsen, K. Stec, and J. Kjeldskov, "Experiences of Personal Sound Technologies," in *Human-Computer Interaction – INTERACT 2021*, ser. Lecture Notes in Computer Science, C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, and K. Inkpen, Eds. Cham: Springer International Publishing, 2021, pp. 523–541.

[40] R. M. Jacobsen, K. Fangel Skov, S. S. Johansen, M. B. Skov, and J. Kjeldskov, "Living with Sound Zones: A Long-term Field Study of Dynamic Sound Zones in a Domestic Context," in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*. Hamburg Germany: ACM, Apr. 2023, pp. 1–14.

[41] R. M. Jacobsen, K. F. Skov, M. B. Skov, and J. Kjeldskov, "Barriers for Domestic Sound Zone Systems: Insights from a Four-Week Field Study," in *Proceedings of the 18th International Audio Mostly Conference*. Edinburgh United Kingdom: ACM, Aug. 2023, pp. 185–192.

[42] K. Fangel Skov, P. Axel Nielsen, and J. Kjeldskov, "Tuning Shared Hospital Spaces: Sound Zones in Healthcare," in *Proceedings of the 18th International Audio Mostly Conference*. Edinburgh United Kingdom: ACM, Aug. 2023, pp. 63–70.

[43] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "PANNs: Large-Scale Pretrained Audio Neural Networks for Audio Pattern Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[44] S. Felipe Contin de Oliveira, F. Aletta, and J. Kang, "Self-rated health implications of noise for open-plan office workers: An overview of the literature," *Building Acoustics*, p. 1351010X231152841, Feb. 2023.

[45] P. Mantello and M.-T. Ho, "Emotional AI and the future of wellbeing in the post-pandemic workplace," *AI & SOCIETY*, Feb. 2023.

[46] A. Berger, A. H. Ambe, A. Soro, D. De Roeck, and M. Brereton, "The Stories People Tell About The Home Through IoT Toolkits," in *Proceedings of the 2019 on Designing Interactive Systems Conference*, ser. DIS '19. New York, NY, USA: Association for Computing Machinery, Jun. 2019, pp. 7–19.

[47] "MAXQDA 2024," VERBI Software, Berlin, Germany, Nov. 2023.

[48] Y. Hou, Q. Ren, H. Zhang, A. Mitchell, F. Aletta, J. Kang, and D. Botteldooren, "AI-based soundscape analysis: Jointly identifying sound sources and predicting annoyance," *The Journal of the Acoustical Society of America*, vol. 154, no. 5, pp. 3145–3157, Nov. 2023.

[49] L. Jia, Z. Zhou, F. Xu, and H. Jin, "Cost-Efficient Continuous Edge Learning for Artificial Intelligence of Things," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7325–7337, May 2022.

[50] K. Saha and S. T. Iqbal, "Focus Time: Effectiveness of Computer Assisted Protected Time for Wellbeing and Work Engagement of Information Workers," in *Proceedings of the 2nd Annual Meeting of the Symposium on Human-Computer Interaction for Work*, ser. CHIWORK '23. New York, NY, USA: Association for Computing Machinery, Sep. 2023.

A Generative Framework for Composition-aware Loop Recommendation In Music Production: Drum2Bass Use Case

Xiaowan Yi (易小琬)[1] and **Mathieu Barthet**[2]

[1,2]*Centre for Digital Music*, **Queen Mary University of London**, London, UK

ABSTRACT

Loops can be defined as short audio items intended to be repeated seamlessly. They have become popular in computer-based music production, particularly in genres such as electronic music and hip hop. A common practice in loop-based music production is to select several loops from a library and play them concurrently. As the producer's loop library expands, a notable challenge emerges in the loop retrieval process: choice overload becomes increasingly prevalent, as navigating and listening to thousands of audio items can be overwhelming, potentially impeding producers' creativity. To facilitate the loop retrieval process, we propose a novel generative framework for loop recommendation that takes into account the composition elements. The framework first generates a best-guess loop conditioned on the current composition elements. The generated best-guess loop then serves both as a serendipitous item for recommendation and an anchor for retrieving existing loops to recommend. We demonstrate this framework with a use case in generating and recommending bass loops based on a seed drum loop (Drum2Bass). We evaluate the retrieval performance of Drum2Bass using computational metrics and provide audio examples of generated and recommended bass loops for listening.

1. INTRODUCTION

In computer-assisted music production, loops generally refer to bar-aligned and structured audio items of short duration [1]. They are widely used in many music genres and in particular, electronic [2] and hip hop music [3]. There are emerging databases to assist composition by providing a wide range of production-ready loops such as Ampify Music [1], Splice [2] and Loopmasters [3]. With such richness of content, musicians of all levels can now access vast amount of composition materials. However, it may impede creativity as it can be daunting to navigate and listen to thousands of audio pieces just to retrieve several loops matching a producer's current composition.

[1] https://ampifymusic.com/
[2] https://www.splice.com/
[3] https://www.loopmasters.com/

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

To this end, we propose a generative framework for composition-aware loop recommendation that aims to facilitate the loop retrieval process during music production. We introduce the concept of *composition-aware recommendation* to describe algorithms that recommend audio content by analysing the present elements within a musical composition. The proposed framework leverages generative models for loop generation, with these generated loops feeding into a novel paradigm for loop recommendation. Given a seed loop in a producer's current composition, the framework firstly generates a loop in raw audio waveform to complement the seed loop. This generated best-guess loop, which does not exist in the producer's current loop library, is recommended to the producer as a serendipitous item. Following this, a similarity-based search is carried out across the loop library using the generated loop as an anchor. Loops within the library that are most similar to the generated best-guess loop are subsequently recommended to the producer. The loop generation capability can contribute to building an explainable recommendation framework as it enables producers to listen to and interpret the intermediate step behind the framework's recommendation process. To test this framework, we investigate the particular use case of generating and recommending bass loops given a drum loop, referred to as Drum2Bass. We evaluate Drum2Bass using both retrieval-relevancy-based and diversity-based metrics.

The remainder of this paper is structured as follows. Section 2 discusses related works and Section 3 addresses the problem context for composition-aware loop recommendation. The proposed framework is described in Section 4. Section 5 reports on the Drum2Bass use case, including the dataset, models, training process, and evaluation. Conclusions and future works are presented in Section 6.

2. RELATED WORK

2.1 Composition-aware Content Retrieval for Music Production

Composition-aware content retrieval for music production is a relatively under-explored area where previous works focused on three types of music content (one-shot drum samples [4], stems [5], and loops [6, 7]) and mainly adopted classification or joint embedding method. Classification methods may use a binary classifier to determine if retrieved music content (like a percussion stem [5]) is a good match for the given composition context (e.g. a seed vocal stem [5]). The music content recommendations are ordered according to the classification scores. Following a

joint embedding method, music content and corresponding composition context are mapped into a learned latent space where being closer according to some metrics indicates a better match. The music content recommendations are ordered by ranking the distance computed between the latent representations. SampleMatch [4] tackled one-shot drum sample retrieval problem using a joint embedding method. Neural Loop Combiner [6] worked on finding good combinations of loops extracted from hip hop music. The authors experimented with both methods and found that the classification method outperformed the joint embedding method in a subjective listening test. Works by Huang et al. [5] modelled stem retrieval for mashup creation via a classification method.

2.2 Variational Autoencoder in Music Content Generation

Variational Autoencoder (VAE) [8] and its variations, which typically comprise an encoder and a decoder, have been adopted in recent works related to music content generation [9]. The VAE family has exhibited strong representation learning capability and the latent representations produced by the encoder are utilised across a range of creative applications. RAVE [9] adopted VAE for modelling raw audio waveforms and obtained compact latent representations by post-training analysis. It demonstrated real-time audio synthesis and timbre transfer capabilities due to the compactness of the latent representation. SoundStream [10] extended VAE with residual vector quantization for producing discrete latent representations from raw audio waveforms. It was adopted in MusicLM [11], a recent text-conditioned music generation model, where the learned discrete latent representations of audio content were used as modelling targets (referred to as acoustic tokens). MusicVAE [12] proposed the use of a hierarchical decoder in the VAE architecture for modelling long-term sequence of music notes. MusicVAE was used to generate interpolations between loop notes in the symbolic domain.

3. COMPOSITION-AWARE LOOP RECOMMENDATION

3.1 Problem Definition

In the context of composition-aware loop recommendation, we introduce the following definitions:

- Loop: A loop is an audio item that can be seamlessly repeated. Loops can feature one or multiple instruments (e.g. bass loop, beat, etc.). While instrumentation varies across loop providers, an example of popular set for electronic music is : drum, bass, melodic, harmonic, vocal. In order to focus the problem, we constrain each loop to have one and only one instrument $Inst_j$.

- Loop pack: Loops are often grouped in packs, which are also referred to as sample packs. Loops within one pack typically have the same key, tempo and genre. Loop packs often share a similar aesthetic within a pack, designed by the provider to sound good when played together right out of the box.

- Loop library L: A loop library L is represented as a set of N different loops $\{l_1, l_2, ..., l_N\}$. These loops may or may not be grouped in loop packs.

- Loop-based music composition: We propose to narrow down our scope to the process of selecting and retrieving loops from a producer's loop library, where retrieved loops get played concurrently to other loops in a music section.

- Composition-aware loop recommendation: Given a loop l_m of instrumentation $Inst_j$ in the current composition, a composition-aware loop recommender aims to produce a ranked list of loops from a different instrument $Inst_{k \neq j}$ specified by the producer. Loops in the recommendation list are ranked based on their predicted degree of music compatibility with the seed loop. This is different than predicting musical similarity to a seed item, as often used in music recommendation for listening.

3.2 Challenges

We identify three main challenges for composition-aware loop recommendation. Firstly, there is a lack of user-item (producer-loop) interaction data in music production software; this data would be crucial to support collaborative filtering prevalent in traditional recommendation systems. To the best of our knowledge, there are no public datasets providing metadata on loop combinations for a varied set of producers, compositions, and genres. Consequently, we propose to adopt a content-based recommendation approach, which circumvent the requirement for user-item interaction data. Audio content-based recommendation usually relies on extracted or learned audio items representations; these are used to infer similarities between items and rank them [13]. Unlike traditional methods that search for similar items, we seek compatible items that complement the seed item. This poses a second research challenge as compatibility-based audio content retrieval is an understudied problem. Both the classification and joint embedding methods require negative data, in our case incompatible loop pairs (loops that do not sound good when being played together). With a lack of datasets providing both compatible and incompatible loops, rule-based strategies (e.g. juxtaposing incompatible keys, tempos and beat phrases in the work by Huang et al. [5] or random associations) are often required for assembling incompatible data from datasets with compatible data. It is shown in the work by Chen et al. [6] that incompatible data mining strategy itself is a system design variable. In their study, the classification method and the joint embedding method exhibited different performances across various data mining strategies, and no optimal strategy was identified. A question remains whether it is possible to develop a loop recommendation framework that can differentiate loops based on compatibility without relying on pre-defined incompatible loop pairs. Lastly, a loop recommendation framework

supports a creative process where properties such as user control over recommendation, serendipity and diversity are sometimes more desirable than metrical accuracy and precision [14]. This poses challenges for the design of the recommendation framework as well as the evaluation methods.

4. PROPOSED FRAMEWORK

We decompose composition-aware loop recommendation into two subproblems: generative modelling and loop pair compatibility modelling. To this end, we propose a generative framework that comprises two training phases with each phase addressing one subproblem.

4.1 Two Training Phases

4.1.1 Phase 1: generative modelling

The first phase is a generative modelling phase where latent-variable generative models are trained to reconstruct loops in raw audio waveforms. There is one generative model trained for each instrument loop category (e.g. one model for drum loops and another model for bass loops), as shown in Fig 1. The artificial neural network architecture we use is the Variational Autoencoder (VAE). The goal of the first phase is to learn compact representations, referred to as embeddings, for each instrument loop and to ensure that such embeddings are capable of generation, i.e. having enough information for the instrument-specific decoder to generate raw audio waveforms. It is worth noting that the data requirement for the first phase is just a loop dataset that has instrument labels for each loop.

4.1.2 Phase 2: mapping compatible loop pair embeddings

Once the first phase of training is done, both instrument loop VAE models are frozen. The second phase is where we try to directly model the mapping relation between compatible loop pair's embeddings, as shown in Fig 1. To this end, we train an extra module to map the different instrument loop embeddings to a shared latent space where proximity indicates compatibility. Compatibility-based recommendation is then reduced to searching for loops that are close to each other in the shared latent space. One example of such module is to train a Multilayer Perceptron (MLP) to, for instance, predict the compatible bass loop embedding given a drum loop embedding. Because the training objective is to directly predict the target compatible loop embedding which is neither a classification score nor contrastive loss, it bypasses the need for incompatible loop pairs mining. Hence, the data requirement for the second phase is a loop dataset that has compatible loop pairs (loops that sound good when played concurrently). Due to the lack of publicly available compatible loop pair datasets, we propose two strategies for creating such datasets. One way is to extract pairs from loop library that has loop-pack structure. Loops of different instruments can potentially be considered compatible if they belong to the same loop pack and share the same key, tempo and genre. This strategy is used in our experiment. Another way is to extract pairs from multitrack music datasets. We can obtain loops by running songs through a loop extraction algorithm [15]. Pairs of loops that originate from the same song section are assumed to be compatible. Neural Loop Combiner [6] adopted this data extraction strategy.

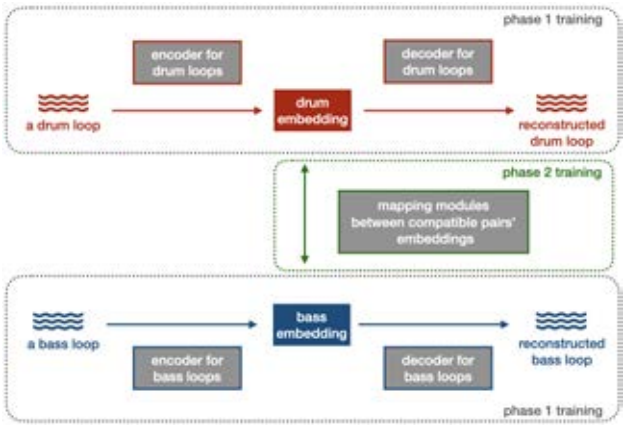


Figure 1. The proposed framework's two training phases. Phase 1: two VAEs are separately trained for each instrument. Phase 2: a mapping module is trained to predict compatible loop's embedding.

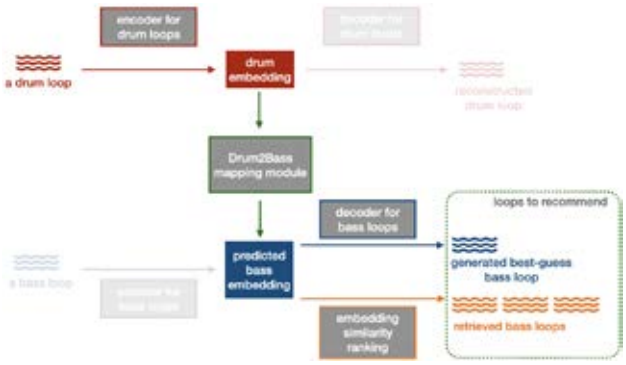


Figure 2. The framework deployment with an example of recommending compatible bass loops from a drum loop through the Drum2Bass mapping module.

4.2 Model Deployment

During deployment, all loop embeddings, which are encoded by the corresponding instrument-specific encoders, can be pre-calculated and stored. Given a seed loop in the composition and a target instrument, the framework retrieves the loop's stored embedding which is then passed into the corresponding latent mapping module that outputs the predicted compatible loop's embedding. The predicted compatible loop embedding can be fed into the decoder for generating a best-guess loop. It can also be used as a recommendation query for retrieving similar loops from the loop library. An example pipeline is shown in Fig 2. It is worth noting that the proposed framework affords several other applications that are specific to music creation. The trained instrument-specific VAE models in phase 1 en-

able timbre transfer and latent interpolation of loops. Timbre transfer can be done by encoding and decoding a loop from instrument $Inst_k$ using the VAE model trained with loops of $Inst_{j \neq k}$, which is similar to the RAVE application demonstration. Latent interpolation can be done by interpolating and decoding the embeddings of two loops of the same instrument. This method creatively combines the essence of two loops to generate new ones, similar to how MusicVAE works.

5. EXPERIMENTS: DRUM2BASS USE CASE

In this work, we address the specific case of recommending a bass loop based on a seed drum loop, referred to as Drum2Bass.

5.1 Dataset

We use a proprietary techno and house music loop library of 44.1 kHz sample rate from Ampify Music. It has 117 loop packs comprised of 846 drum loops and 613 bass loops. Each pack has at least one drum loop and one bass loop, and some packs contain more drum loops than bass loops. We extract 846 compatible loop pairs by matching each drum loop with one bass loop from the same pack. We then split the dataset into training/validation/test sets using the ratios 0.85/0.05/0.1. The split is done at the loop pack level so that loops from the same pack are not found in different sets.

5.2 Models

For the first phase, we choose RAVE [9] as our VAE model implementation. We separately train one RAVE model on drum loops (*DrumRAVE*) and another one on bass loops (*BassRAVE*). The training objective for each model comprises a reconstruction loss term and a prior regularisation loss term:

$$L_{RAVE} = E_{\hat{x} \sim p(x|z)}[S(x, \hat{x})] + \beta * D_{KL}[q_\phi(z|x)||p(z)], \quad (1)$$

where x is a loop from the dataset in the audio waveform, z is the latent variable with a prior distribution $p(z)$. $q_\phi(z|x)$ is the posterior distribution parameterised by the RAVE encoder and $p(x|z)$ is the conditional likelihood distribution parameterised by the decoder. $S(x, \hat{x})$ denotes the multi-scale spectral distance between the original and reconstructed waveform. D_{KL} denotes the Kullback-Leibler divergence. After training, both models' encoders and decoders are frozen before proceeding to the second phase.

For the latent mapping module in the second phase, we start with an MLP of four fully connected layers that maps drum embeddings to compatible bass embeddings. The training objective for the MLP is the L2 distance in the latent space, calculated between the predicted and the groundtruth compatible bass loop embeddings:

$$L_{latent} = ||\hat{y} - BassRAVE_{enc}(y)||_2, \quad (2)$$

with

$$\hat{y} = MLP(DrumRAVE_{enc}(x)), \quad (3)$$

where y is the groundtruth compatible bass loop (in the audio domain) for seed drum loop x, and $\hat{y}$ is the predicted compatible bass loop embedding. $BassRAVE_{enc}$ and $DrumRAVE_{enc}$ denote the RAVE encoders respectively. We also experiment with the L2 distance computed in the decoded space, which is calculated after running embeddings through the frozen BassRAVE decoder:

$$L_{dec} = ||BassRAVE_{dec}(\hat{y}) - y_{dec}||_2, \quad (4)$$

where

$$y_{dec} = BassRAVE_{dec}(BassRAVE_{enc}(y)). \quad (5)$$

We denote the MLP variant with loss computed in the latent space (i.e. without being decoded) by *MLP-Latent* and the MLP variant with loss computed in the decoded space by *MLP-Dec*. Our hypothesis is that *MLP-Latent* will yield better compatible loop retrieval performance as the embedding similarity ranking deploys distance measured in the latent space instead of the decoded space. *MLP-Dec* may produce better loop generation quality as it optimizes towards raw audio waveform distance.

5.3 Training

For training the RAVE models in phase 1, audio segments of a fixed length of 65536 sample points are randomly selected from each loop with zero-padding. We train DrumRAVE and BassRAVE each for $3,000,000$ training steps with batch size 8 and start with a learning rate 0.0001 using ADAM optimizer. For training the MLP in phase 2, we extract a 4-second window from each loop pair to compute embeddings with tiling where applicable. We choose to keep loops at their original tempo so no time-stretching is applied. As the slowest tempo from the loop library is 60 BPM, 4-second window ensures there is at least 4 beats of content taken into account. The input and output size of the MLP, which is the RAVE embedding size, is $(128, 87)$ for each 4-second loop where 128 is the feature resolution and 87 is the temporal resolution. We train *MLP-Latent* and *MLP-Dec* each for $50,000$ training steps with batch size 32 and start with a learning rate 0.0001 using ADAM optimizer.

5.4 Results and Discussion

The evaluation of the Drum2Bass framework is divided into two parts: one focuses on loop generation performance, and the other assesses loop recommendation performance with computational evaluation detailed in Section 5.4.1. For loop generation performance, we invite readers to listen to the audio examples here[4]. Overall, the generated bass loops adhere to the seed drum loop's tempo without being explicitly conditioned on tempo information. The generated loops exhibit diversity across different seed drum loops, and there are interesting rhythmic patterns generated such as syncopation. The generation quality still needs improvement, though we have not performed extensive hyper-parameter tuning in this experiment.

[4] https://bit.ly/generative-loop-recommender-audio-examples

Variant	R_{mn}	R_{md}	ILD@5	ILD@10
MLP-Latent	**0.394**	**0.306**	0.623	0.640
MLP-Dec	0.524	0.473	0.626	0.638

Table 1. Computational Evaluation Results

5.4.1 Computational Evaluation

Here we report the evaluation results regarding the loop recommendation performance, with computational metrics shown in Table 1. Given a seed drum loop i, we create a recommendation list of bass loops by ranking them according to the cosine similarities between their embeddings and the generated best-guess bass embedding. We then assess the quality of the recommendation list using both rank-based metrics and a diversity-based metric. Specifically, we propose to use Mean Normalized Rank R_{mn} and Median Normalized Rank R_{md} as rank-based metrics:

$$R_{mn} = \frac{1}{N_{drum}} \sum_{i=0}^{N_{drum}} \frac{rank_i}{N_{bass}}, \tag{6}$$

$$R_{md} = median\{\frac{rank_i}{N_{bass}} \,|i = 1, 2, ..., N_{drum}\}, \tag{7}$$

where N_{drum} and N_{bass} denote the number of drum loops and bass loops, respectively. We use $rank_i$ to denote the rank of the groundtruth compatible bass loop in the ranked recommendation list for the seed drum loop i. Dividing $rank_i$ by N_{bass} normalizes it into $(0, 1]$. R_{mn} represents the mean of all normalized ranks, while R_{md} represents the median, effectively ignoring potential outliers. The closer the normalized rank is to 0, the better the framework performs in ranking the most compatible bass loop higher in the recommendation list among other bass loops, thereby alleviating choice overload for music producers. These rank-based metrics were also adopted in related works, including SampleMatch [4] and Neural Loop Combiner [6], for the same reason. In response to the challenge mentioned in Section 3.2 regarding the assessment of loop recommendation frameworks within the context of creative applications, we introduce a diversity-based evaluation. We use an inter-list diversity metric (ILD) which considers the uniqueness of different drum loops' bass loop recommendation lists:

$$ILD@L = \frac{1}{\binom{N_{drum}}{2}} \sum_{i \neq j}^{N_{drum}} 1 - \frac{q_{ij}(L)}{L}, \tag{8}$$

where $q_{ij}(L)$ is the number of common bass loops in the top L places of any two drum loops' recommendation lists. $\binom{N_{drum}}{2}$ is the normalization term denoting the total number of combinations when choosing two drum loops from N_{drum} loops. A higher value of $ILD@L$ generally indicates greater diversity among the top L bass loop recommendations for any two drum loops.

As can be seen in Table 1, the best rank-based evaluation result is achieved by the variant *MLP-Latent*. It outperforms the random sampling baseline (which has a normalized rank of 0.5) and *MLP-Dec* in both R_{mn} and R_{md} which is in line with our initial hypothesis. We conjecture that this is because the loss measured in the decoded space may not directly agree with the loss measured in the latent space. The final recommendation item ranking is solely based on the distance measured in the latent space which favors *MLP-Latent*. The results on $ILD@5$ and $ILD@10$ indicate that *MLP-Latent* and *MLP-Dec* exhibit similar diversity-based recommendation performance. On average, more than half of the bass loops in one seed drum loop's recommendation list differ from those in any another seed loop's recommendation list. Notably, $ILD@5$ values of 0.623 (*MLP-Latent*) and 0.626 (*MLP-Dec*) imply that, on average, more than three out of the top five recommended loops are different between any two recommendation lists.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the challenges of composition-aware loop recommendation and proposed a novel two-phase generative framework to address these challenges. We demonstrated the framework through the Drum2Bass use case, which generates and recommends bass loops based on a seed drum loop. Our experiments with Drum2Bass highlighted the framework's capabilities in loop generation, loop compatibility modeling without relying on negative (incompatible) data mining, and loop recommendation with inter-list diversity. We investigated two phase-2 loss variants in Drum2Bass, and the computational evaluation results showed that the loss computed in the latent space achieved better retrieval performance than the loss computed in the decoded space. Both variants had high inter-list recommendation diversity. However, these two computational metrics might be partial in evaluating a recommendation framework for music production. We will extend the evaluation with a user study in the future to better reflect the system's efficacy during actual music production. As there is still some headroom for improvement in loop generation quality and retrieval performance, we will refine Drum2Bass by experimenting with other state-of-the-art techniques including adding latent code quantization for phase 1 training and adopting models with greater expressiveness (e.g. a decoder-only transformer architecture) for phase 2 training. Additionally, we want to extend our experiment in the future to incorporate public dataset such as the Freesound Loop Dataset [16] and establish benchmarks for the composition-aware loop recommendation challenge.

Acknowledgments

This work is funded by UK Research and Innovation [grant number EP/S022694/1] and Focusrite. Our special thanks are extended to Focusrite Ampify Music team for their technical and dataset support.

7. REFERENCES

[1] G. Stillar, "Loops as genre resources," *Folia Linguistica*, vol. 39, no. 1-2, pp. 197–212, 2005.

[Online]. Available: https://doi.org/10.1515/flin.2005.39.1-2.197

[2] M. J. Butler, *Unlocking the groove: Rhythm, meter, and musical design in electronic dance music*. Indiana University Press, 2006.

[3] D. Charnas, *Dilla time: The life and afterlife of J Dilla, the hip-hop producer who reinvented rhythm*. MCD, 2022.

[4] S. Lattner, "Samplematch: Drum sample retrieval by musical context," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, pp. 781–788. [Online]. Available: https://archives.ismir.net/ismir2022/paper/000094.pdf

[5] J. Huang, J. Wang, J. B. L. Smith, X. Song, and Y. Wang, "Modeling the compatibility of stem tracks to generate music mashups," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. AAAI Press, 2021, pp. 187–195. [Online]. Available: https://doi.org/10.1609/aaai.v35i1.16092

[6] B. Chen, J. B. L. Smith, and Y. Yang, "Neural loop combiner: Neural network models for assessing the compatibility of loops," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, pp. 424–431. [Online]. Available: http://archives.ismir.net/ismir2020/paper/000225.pdf

[7] F. Bruford, O. Lartillot, S. McDonald, and M. B. Sandler, "Multidimensional similarity modelling of complex drum loops using the groovetoolbox," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 263–270. [Online]. Available: http://archives.ismir.net/ismir2020/paper/000211.pdf

[8] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. [Online]. Available: http://arxiv.org/abs/1312.6114

[9] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," *CoRR*, vol. abs/2111.05011, 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[10] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 495–507, 2021. [Online]. Available: https://doi.org/10.1109/TASLP.2021.3129994

[11] A. Agostinelli, T. I. Denk, Z. Borsos, J. H. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. H. Frank, "Musiclm: Generating music from text," *CoRR*, vol. abs/2301.11325, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2301.11325

[12] A. Roberts, J. H. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A hierarchical latent vector model for learning long-term structure in music," in *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, ser. Proceedings of Machine Learning Research, vol. 80. PMLR, 2018, pp. 4361–4370. [Online]. Available: http://proceedings.mlr.press/v80/roberts18a.html

[13] M. Schedl, H. Zamani, C.-W. Chen, Y. Deldjoo, and M. Elahi, "Current challenges and visions in music recommender systems research," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 2, pp. 95–116, 2018. [Online]. Available: https://doi.org/10.1007/s13735-018-0154-2

[14] M. Schedl, P. Knees, and F. Gouyon, "New paths in music recommender systems research," in *Proceedings of the Eleventh ACM Conference on Recommender Systems, RecSys 2017, Como, Italy, August 27-31, 2017*. ACM, 2017, pp. 392–393. [Online]. Available: https://doi.org/10.1145/3109859.3109934

[15] J. B. Smith and M. Goto, "Nonnegative tensor factorization for source separation of loops in audio," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2018, Calgary, AB, Canada, April 15-20, 2018*. IEEE, 2018, pp. 171–175. [Online]. Available: https://doi.org/10.1109/ICASSP.2018.8461876

[16] A. Ramires, F. Font, D. Bogdanov, J. B. L. Smith, Y. Yang, J. Ching, B. Chen, Y. Wu, W. Hsu, and X. Serra, "The freesound loop dataset and annotation tool," in *Proceedings of the 21th International Society for Music Information Retrieval Conference, ISMIR 2020, Montreal, Canada, October 11-16, 2020*, 2020, pp. 287–294. [Online]. Available: http://archives.ismir.net/ismir2020/paper/000241.pdf

PAPER & POSTER /DEMO SESSION 3

/ Vibrotactile Memory: a Case Study of Timbre Perception Training in Children with Cochlear Implants Using a Video Game

/ Assessing Musical Preferences of Children on the Autistic Spectrum: Implications for Therapy

/ An Accessible Software Interface for Collaborative Music Performance

/ Singing for Anxiety: Exploring the Potentials of Singing in Immersive Virtual Reality to Help Children with Social Anxiety

/ From MIDI to Rich Tablatures: an Automatic Generative System Incorporating Lead Guitarists' Fingering and Stylistic Choices

/ I Got Rhythm, so Follow Me More: Modeling Score-dependent Timing Synchronization in a Piano Duet

/ Simulating Piano Performance Mistakes for Music Learning

/ Tablature Generation from Lead Sheets for Finger-style Solo Guitar

/ Exploring Uncharted Soundscapes: Innovative Interaction and Mapping Techniques with the Bodyharp and the Teinophon

VIBROTACTILE MEMORY: A CASE STUDY OF TIMBRE PERCEPTION TRAINING IN CHILDREN WITH COCHLEAR IMPLANTS USING A VIDEO GAME

Francesco GANIS (frga@create.aau.dk) (0000-0001-8146-1156)[1],
Ali ADJORLU (adj@create.aau.dk) (0000-0001-5414-8631)[1],
Lone M. PERCY-SMITH (lone.percy-smith@regionh.dk) (0000-0002-5904-3879)[2],
Cecilia F. SAMAR (cecilia.fernandez.samar@regionh.dk)[2], and
Stefania SERAFIN (sts@create.aau.dk) (0000-0001-6971-1132)[1]

[1]*Architecture Design and Media Technology*, **Aalborg University**, A. C. Meyers Vænge 15, 2450 Copenhagen, Denmark
[2]*Hearing and Balance Center, Ear, Nose and Throat (ENT) and Audiology Clinic*, **Rigshospitalet - Copenhagen University Hospital**, Inge Lehmanns Vej 8, 2100 Copenhagen, Denmark

ABSTRACT

This paper presents a preliminary study investigating the effectiveness of a computer game designed for enhancing timbre perception in children with cochlear implants. The game focuses on instrument recognition and incorporates vibrotactile feedback through the PlayStation 5 DualSense controller. A case study was conducted involving a 7-year-old participant diagnosed with hearing loss due to auditory neuropathy who uses both a cochlear implant and a hearing aid. Training sessions were conducted over a two-week period, with assessments performed before and after the training thanks to a modified version of the Timbre Perception Test. Despite incorporating vibrotactile feedback and tracking participant progress, the study did not observe significant improvements in timbre perception. Factors such as the short duration of the training period, the difficulty of the game, and potential boredom may have influenced the outcomes. The study highlights the need for further research to refine the training intervention, including extending the duration, enhancing engagement, and conducting larger-scale studies with diverse participant groups.

1. INTRODUCTION

Cochlear implants (CIs) are surgically implanted neural prostheses designed to restore hearing in people with profound hearing loss (HL) [1]. Until recently, researchers and companies focused mainly on optimizing these devices for speech, obtaining good results in reestablishing language comprehension [2]. These devices, however, still present challenges for music appreciation. Music, in fact, is a highly complex phenomenon with many factors that can influence its enjoyment and access for hearing aids (HA) and CI users [3–5]. The hardware, software, and physical limitations of CIs result in a reduced frequency range, incapability of distinguishing sounds' consonance and dissonance, limited dynamic range, and difficulty in recognizing instruments (timbre) [6]. Consequently, music enjoyment and the emotions that it can elicit are directly affected, most often resulting in limited fruition. In addition, musical background, degree of success of the surgery, as well as the presence of other pathologies, have an effect on the hearing experience, making every user's sound experience unique.

These general considerations apply to young recipients too, but some additional aspects have to be considered when children are taken into account. Early cochlear implantation has been shown to yield long-term positive effects, providing children with speech capabilities comparable to their normal hearing (NH) peers before the age of four [7]. After age four, young recipients of CIs develop speech slower than NH individuals, making personalization of learning goals a key aspect of training [8]. Ad-hoc training is even more important when the child suffers from auditory neuropathy (AN), a pathology that implies disrupted neural activity that impairs timing perception, low frequencies' perception, temporal integration, gap detection, sound localization among other things [9]. However, children with HL diagnosed in recent years that are fitted with HA and CI before one year of age, may perform differently from previous generations.

Emerging evidence shows that music training improves music perception skills and experience [6, 10, 11], together with psychosocial well being and quality of life [12]. One of the novel techniques to possibly improve musical listening performance is the use of vibrotactile feedback [13]. Vibrotactile feedback refers to the mappings between audio and vibrations that can help to emphasize specific features such as pitch [14, 15]. Another common approach that improves learning experiences for children is the gamification of the training activities [16, 17]. This strategy can guide youngsters with hearing loss towards improving their musical skills [18], especially when HL is due to auditory neuropathy.

Gamification has demonstrated its effectiveness as a tool for enhancing engagement and motivation across various domains, ranging from business to marketing [19]. It has also shown positive impacts on the learning process, improving engagement and knowledge acquisition [20]. Consequently, gamification appears to be a promising approach

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

for boosting participation in training activities among individuals with hearing impairments. However, it remains a relatively new area of inquiry that has yet to be thoroughly explored [21].

In this paper, we introduce a computer game for musical training of children with cochlear implants, with the aim to improve timbre perception through instrument recognition. The game presents different levels of difficulty and features vibrotactile feedback conveyed through the PlayStation 5 controller DualSense [1] . We report a preliminary case study [22] to collect information about user interaction and efficacy to train the timbre perception.

In Section 2 we present the methodology applied for this case study, including the process and design choices made to create the training video game; in Section 3 we showcase the data retrieved from this experiment and finally in Section 4 we analyze the results and draw some considerations related to the whole case study.

2. METHODOLOGY

2.1 Participant

The participant was 7 years old at the time of the experiment. At the age of 4, she received a cochlear implant in her left ear, and she uses a hearing aid in her right ear (bimodal). She has been diagnosed with AN, a condition that can pose challenges in the transmission of signals from the auditory nerve to the brain's auditory cortex. No assessment of the actual hearing capabilities of the subject has been performed for the scope of this project. We did not characterize the user's perception of vibrotactile stimulation; given the participant's young age, we assumed they have ideal sensitivity capabilities, as most degradation occurs with aging or due to traumas [23] (e.g., heavy-machinery use). The participant has been recruited through the Center for Hearing and Balance from the Rigshospitalet of Copenhagen with approval from the parents and the clinicians.

2.2 Design

In the following sections, we present the project's design process and the experiment.

2.2.1 Participatory Design

We applied the principles of *participatory design* [24, 25] involving the end user in the process to open a dialogue to gather ideas and feedback throughout the development. The child, together with the parents and the audio-verbal therapist contributed to the process through an official meeting and some more informal gatherings. During the first meeting, we collected information about the preferred video games from the child as well as clinical practices and training tools already available. In the follow-ups, we mainly included the clinician to receive feedback about the experiment as well as the interface.

[1] Sony Playstation DualSense Controller `https://www.playstation.com/en-dk/accessories/dualsense-wireless-controller/` — Last access May 22, 2024

2.2.2 Experiment Design

We designed an experiment that features a video game aimed at training children to improve their timbre perception through listening to musical instruments. The experiment includes an initial assessment of timbre perception, a training period using the video game, and a final assessment of timbre perception. In addition to sound feedback, vibrotactile feedback was incorporated to convey musical information through the sense of touch. The hardware setup comprised a laptop equipped with loudspeakers and a controller to provide both vibrotactile feedback and input capabilities. In Figure 1 the simple setup is depicted.

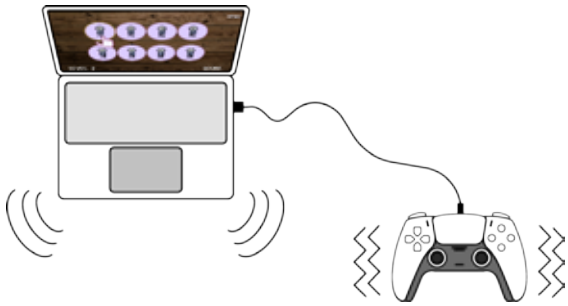


Figure 1. Setup for the training experience.

The experiment extended over a two-week period, with the participant conducting the training activity at home three times a week under the supervision of her parents. The training tool systematically tracked the progress of each participant, facilitating a comprehensive analysis of their performance. Ultimately, we conducted interviews with both the child and their parents at the conclusion of the training period to gain insights into their overall experience and gather feedback on potential enhancements to the prototype.

To assess the efficacy of our intervention, a modified version of the Timbre Perception Test (TPT) [26] was administered both before and after the training in the clinic together with the researchers and the audiologist. The modifications involved scaling the parameters to create a more pronounced difference in the controls, ensuring they were easily audible to the participant.

2.3 Assumptions

Given a single participant and the specific design of the test, we did not formulate hypotheses but instead made some assumptions regarding our expectations concerning the training activity that involved the child with AN:

1. Increasing the training duration will lead to a reduction in the required interaction, measured by the number of clicks, with the cards.

2. Incorporating vibrotactile feedback will improve the ability to discriminate between the cards.

2.4 Training Environment

While creating the training game, we drew inspiration from the classic memory card game, where players take turns

flipping pairs of cards to find matching pairs. In our design, we incorporated the same mechanics, but rather than requiring the player to focus on images, we asked them to concentrate on sounds, with particular attention to the timbre of the instrument being played. Once a card is selected, the corresponding sound (and eventually vibration) is played through the laptop's loudspeakers. Only when a matching pair is found, the cards turn, revealing the image of the instrument playing. This approach is borrowed from auditory verbal practice, where sound is provided before vision; thus, the patient is induced to focus first on sound, while visual feedback is given only at a later stage to reinforce the association [8] aiming to enhance both listening capabilities in terms of timbre quality and auditory memory.

We developed the game using the Unity game engine (*ver. 2021.3.11f1*) [27] in a 2D environment. User interaction is made possible through the use of the DualSense controller from Sony. This choice led us to face the problem of feeding the user with a synchronized and coherent vibrotactile signal with the audio. The controller, when connected with a USB cable, behaves as both an input device and vibrotactile feedback. To provide both sound and vibrotactile feedback on independent streams, we opted for controlling the audio and haptic stream through FMOD for Unity [28]. This allows to simultaneously use the integrated sound card of the host laptop and the controller's internal sound card. In Figure 2 we can observe a screenshot of the video game. In the bottom side of the screen, level and scores are reported. From the main menu of the game it is also possible to check the progress of the game and play again the completed levels for further training.

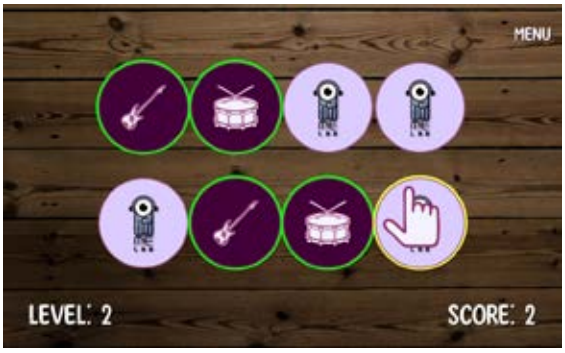


Figure 2. View of the memory game.

2.4.1 Level Generation

The game consists of 16 levels with increasing difficulty. The parameters that vary between the levels include the number and similarity of instruments, as well as the melodies played. Instrument similarity is defined in terms of timbre, where, for instance, a violin is very similar to a viola but remarkably different from a trumpet. Table 1 illustrates how the levels are structured in the game. It's important to note that each level has two different versions—one with haptic feedback and one without. This explains why we report only 8 levels in Table 1 while in the game there are 16. These versions run consecutively, though this detail is not included in the table for the sake of readability.

Level	# Instr.	Similarity	Melody
1.1	4	Different	Different
1.2	5	Different	Different
2.1	4	Different	Same
2.2	5	Different	Same
3.1	4	Similar	Different
3.2	5	Similar	Different
4.1	4	Similar	Same
4.2	5	Similar	Same

Table 1. Level's difficulty structure

In order to prevent any bias, the instrument position and type, and the melodies are randomly assigned to each level and re-calculated at every run.

2.4.2 Data Acquisition

With the aim in mind of tracking the progress of the participant, we implemented a saving system that records the content of each level (i.e., instruments and melodies), the number of clicks per card, and the time it took to complete the level. All this information is stored in separate `.json` files saved in a hidden folder of the host computer's file system to prevent data manipulation from the end user.

2.5 Stimuli and Instruments

To exert the highest degree of control over the musical stimuli present in the experiment, we chose to generate all the recordings from MIDI files. This involved using a set of four AI-generated melodies [2], one ascending and one descending scale, and two famous excerpts (Eine Kleine Nachtmusik and Piano Sonata No. 16 "Sonata Semplice," both composed by Mozart), totaling eight melodies. All recordings are played at 100 BPM and last for two bars, covering a range from A3 to F5 (220 - 698.5 Hz). The MIDI files were employed to feed a set of audio plug-ins based on either sampled or physically modeled instruments to achieve high-quality audio and fidelity. The instruments selected for the experiment are listed in Table 2, grouped by timbre similarity.

Instrument	Category	Instrument	Category
Xylophone	1	Violin	3
Piano	1	Trumpet	4
Guitar	1	Trombone	4
Bass	2	Flute	5
Cello	3	Sax	5
Viola	3	Clarinet	5

Table 2. Instruments categories grouped by timbre similarity

The sound stimuli were generated by loading the MIDI files into Reaper [29], a Digital Audio Workstation (DAW), along with the virtual instruments shown in Table 2. The

[2] Magenta Studio Generator — `https://magenta.tensorflow.org/studio` — Last access May 22, 2024

individual audio files were then exported at 44.1 kHz and 16 bits. All the stimuli were normalized at -12 dB to maintain an equal perceived loudness. These same files were used for both the audio and vibrotactile feedback. The experiment was conducted feeding the vibrotactile actuators with unprocessed audio signal. This choice was driven by the type of training task: since we focus on timbre and musical instrument recognition, we wanted to preserve the frequency spectrum and the envelope of the instruments, following the approach used by Russo *et al.* [30]. In future iterations we will develop other training activities that will include melodic contour identification, and we will use different mapping techniques such as the one used in [15].

2.6 Vibrotactile Input

Humans' mechanoreceptive system is capable of perceiving vibrations up to approximately 1 kHz ([23]) and therefore choosing actuators able to cover this frequency range is crucial. In literature, it is possible to find several prototypes available such as vests, gloves, and furniture ([31]). These solutions are not suitable for our project, since the target group requires rugged devices that are easy to use and feature plug-and-play behavior. Consequently, we opted for a commercially available gaming console controller that children might be familiar with. In addition, these devices can be brought at home and connected to any laptop, giving a good degree of flexibility. Recently, Sony released the PlayStation 5 with the DualSense controller (2020) that features high quality vibrotactile feedback conveyed through voice-coil actuators. The working principle of this technology is similar to the loudspeaker, and thus shares a comparable performance in frequency range.

3. RESULTS

3.1 Video Game's Data

As introduced in Section 2.4.2, we tracked how many times the user interacted with the cards by clicking on them and listening to the sound. In Figure 3, the total number of clicks per session can be seen. The number shown on top of each bar indicates the level in the video game. The bars with the green outline represent the levels with 5 pairs of cards, while the ones without represent 4 pairs of cards. Additionally, there are green and white lines representing the optimal number of moves for 5 and 4 pairs of cards, respectively, where the optimal number represents the expected number of flips (clicks in our case) with a player presenting a perfect memory as demonstrated by Velleman *et al.* [32]. The optimal moves for n pairs of cards are calculated as following:

$$(3 - 2\,ln\,2)\,n + 7/8 - 2\,ln\,2 \approx 1.61 \times n$$
$$n = 4 \rightarrow opt.moves \approx 6.44$$
$$n = 5 \rightarrow opt.moves \approx 8.05$$

In figure 3 we can notice that the interaction between the cards for both levels with 4 and 5 pairs of cards are approximately twice the optimal moves reported in the formulas above.

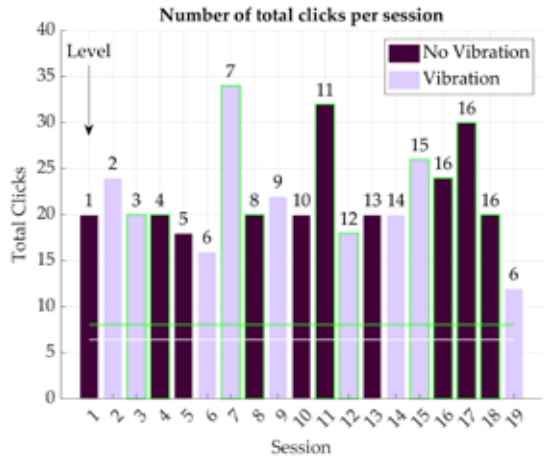


Figure 3. Total amount of clicks on the cards per every level. The levels with green outline are the ones with two extra cards. The white and green lines indicate the optimal moves for 8 and 10 cards respectively.

Observing Figure 4, there is no recognizable patter or trend that could suggest an effect of the vibrotactile feedback or total training time on the average number of clicks per card. The average number of clicks per card among all sessions is 2.42.

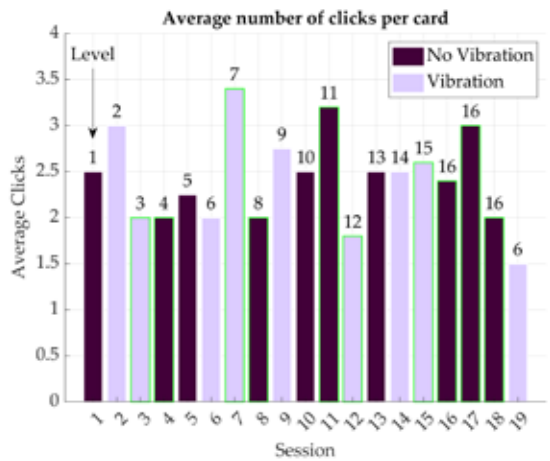


Figure 4. Average amount of clicks on the cards per every level. The levels with green outline are the ones with two extra cards.

In Figure 5 the amount of time elapsed to complete each level is reported. We can observe that the average duration for levels 11 to 16 (excluding the 14th session) is 6 minutes and 8 seconds. This means that participants took between 1 minute and 13 seconds and 1 minute and 32 seconds to find a correct pair, depending on whether the level presented 4 or 5 pairs of cards.

Figure 6 shows the score deviation between the target sound and the answer from the participant. The values represent the number of steps from the slider used in the TPT interface. Running a t-test on the difference between the

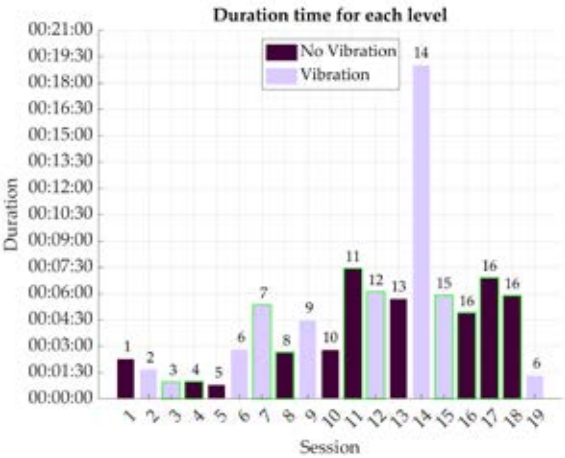


Figure 5. Time elapsed to complete each level.

answer and the target values, we found a p-value of 0.2466 showing no significant difference between the pre- and post-training. For more information we suggest referring to the manuscript [26].

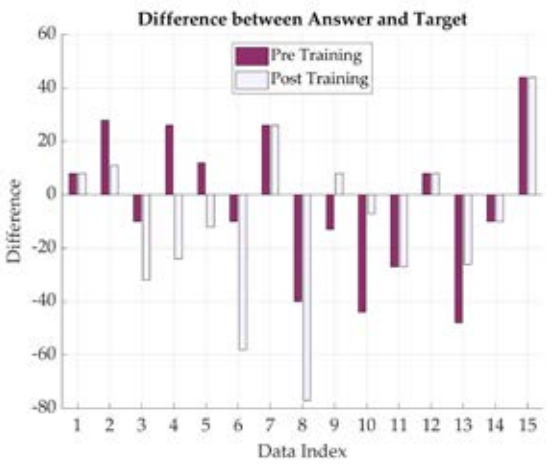


Figure 6. TPT before and after training comparison.

3.2 Interview

To prevent any possible bias, we requested the auditory-verbal therapist to hold a brief interview with both parents and the child in the following session after the training. This approach capitalizes on the established trust between the parties and the familiarity between the child and the therapist. The therapist was given prior instructions to pose targeted questions and guide the discussion to gather relevant information. The conversation was conducted in Danish.

The child reported facing boredom during the training and would not like to repeat the training experience. The parents agree that the game "[...] seemed to be another training". On the other hand, they stated that the game made their child think about the sounds, and she might have learned about the musical instruments. They had to supervise her during all the sessions also because she found a way to cheat, pressing the cards randomly until finding the matching pair. They think it could be suitable for younger recipients too. Finally, they suggest including a more effective rewarding system to improve engagement, and they also think it would be meaningful to be able to listen again to the instruments even after finding the correct pair.

4. DISCUSSION

The findings of this study suggest that the video game intervention designed to enhance timbre perception in children did not produce the expected improvements in participant's performance on the Timbre Perception Test (TPT). Despite the incorporation of vibrotactile feedback, which was intended to provide additional sensory information and enhance participant's ability to discriminate between sounds, the intervention did not yield significant results. This raises questions about the effectiveness of the training approach and potential factors influencing its outcomes.

Several factors may have contributed to the lack of observed improvement in timbre perception following the training intervention. Firstly, the relatively short duration of the training period, conducted over a two-week period with sessions held three times a week, may not have been sufficient for the participants to develop meaningful improvements in their auditory skills. Secondly, this duration did not allow for gathering enough data; we aimed to have more repetitions of the same levels to perform significant data analysis. The difficulty of the game and potential boredom with its repetitive nature could have impacted engagement and motivation, thereby limiting the effectiveness of the intervention as well as the production of data. This is a crucial aspect in the case study as it demonstrates how the current design includes gamification elements in the training activity without being successful. The child, in fact, reported that she usually plays mainstream video games with her friends or alone, and thus we can assume that these games are her comparison term when talking about entertaining activities. The current design features basic gamification mechanisms and, as stated by the parents, could maybe be suitable for younger children in the age range between 4 and 5.

Another point of discussion is the difficulty of the task: from Figure 5, we can observe that the average duration for levels 11 to 16 (excluding the 14th session) is 6 minutes and 8 seconds. This means that participants took between 1 minute and 13 seconds and 1 minute and 32 seconds to find a correct pair, depending on whether the level presented 4 or 5 pairs of cards. Thus, we can conclude that these levels were significantly more difficult than the first ten, indicating a greater challenge in finding similar instruments. This is congruent with our expectations since the harder level presents instruments with similar timbre, thus more challenging to distinguish. In the future, we will reconsider the complexity of the task, making the difficulty increase more gradually and extending the number of levels to follow a gentler learning curve. Level 14 can be considered an outlier, as the completion time is three times longer than the closest levels in difficulty. It can be assumed that the session was not closed before a break, leaving the timer

running until the resumption of the session.

From the graphs in the Results Section 3, no effect of vibrotactile feedback on both the average number of clicks and duration time is apparent. One hypothesis could be that the activity was extremely demanding for the auditory channel, leaving no space for attention on the tactile one. Another possibility is that the child did not fully grasp the connection between the feedback and the sound due to their young age. Alternatively, it could be interpreted that the vibrotactile feedback is not capable of conveying relevant information for musical timbre. This interpretation contrasts with the findings of Russo *et al.* [30]. However, it's worth noting that their study utilized a completely different setup (two voice-coils on a chair), presented the vibrotactile stimuli without sound and involved an older group of participants.

The TPT allowed us to measure the timbre perception of the participant but did not test the learning experience in terms of recognition of the specific instruments presented in the study. In fact, the parents of the child reported that the child might have learned about the instruments thanks to the training video game, but this has not been proven.

The findings of this study underscore the need for further research to refine and optimize the training intervention for timbre perception. Future iterations could focus on extending the duration of the training period to allow for more comprehensive skill development, but only with variations in the game design to increase engagement and motivation. For instance, the game could be part of a suite of different mini-games aimed at training musical skills of children with AN. The design will be put under revision to improve both the dynamics and the interaction with the aim of making the game more appealing to the target group.

It is important to acknowledge the limitations of this study, including the small sample size and the lack of a control group for comparison. As mentioned at the beginning of this paper, the study has to be considered as a case study that requires further validation. Allowing the participant to train at home ensured that the activity could be performed in an ecologically valid environment. However, this approach involved some compromises, such as the absence of researcher supervision, leading to uncertainty about the participant's interaction with the controller. Although the controller's design suggests a specific method of interaction and handhold position, we could not guarantee that the participant held the device optimally. Therefore, the participant's grip on the device might have affected the vibrotactile feedback contribution. Additionally, the use of a single participant group and the specific design of the training intervention may limit the generalizability of the findings. Future research should aim to address these limitations by conducting larger-scale studies with diverse participant groups and incorporating control conditions to more rigorously evaluate the effectiveness of the intervention.

5. CONCLUSIONS

In conclusion, while the initial findings of this study did not demonstrate significant improvements in timbre perception following the training intervention, they provide valuable insights into the challenges and considerations involved in developing effective auditory training interventions for children. By addressing the limitations and exploring alternative approaches, future research has the potential to enhance our understanding of auditory perception and improve the effectiveness of interventions aimed at developing auditory skills in children with CI and AN.

Acknowledgments

The authors would like to thank the participant and her family for taking part in experiment.

6. REFERENCES

[1] O. Macherey and R. P. Carlyon, "Cochlear implants," *Current Biology*, vol. 24, no. 18, pp. R878–R884, Sep. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0960982214007696

[2] Fan-Gang Zeng, S. Rebscher, W. Harrison, Xiaoan Sun, and Haihong Feng, "Cochlear Implants: System Design, Integration, and Evaluation," *IEEE Reviews in Biomedical Engineering*, vol. 1, pp. 115–142, 2008. [Online]. Available: http://ieeexplore.ieee.org/document/4664429/

[3] V. Looi, H. Mcdermott, C. M. McKay, and L. Hickson, "The effect of cochlear implantation on music perception by adults with usable pre-operative acoustic hearing," *International Journal of Audiology*, vol. 47, pp. 257 – 268, 2008.

[4] W. R. Drennan, J. J. Oleson, K. Gfeller, J. Crosson, V. D. Driscoll, J. H. Won, E. S. Anderson, and J. T. Rubinstein, "Clinical evaluation of music perception, appraisal and experience in cochlear implant users," *International Journal of Audiology*, vol. 54, no. 2, pp. 114–123, Feb. 2015. [Online]. Available: http://www.tandfonline.com/doi/full/10.3109/14992027.2014.948219

[5] C. J. Limb and A. T. Roy, "Technological, biological, and acoustical constraints to music perception in cochlear implant users," *Hearing Research*, vol. 308, pp. 13–26, Feb. 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378595513001020

[6] N. T. Jiam, M. T. Caldwell, and C. J. Limb, "What Does Music Sound Like for a Cochlear Implant User?" *Otology & Neurotology*, vol. 38, no. 8, pp. e240–e247, Sep. 2017. [Online]. Available: https://journals.lww.com/00129492-201709000-00033

[7] O. B. Wie, J. V. K. Torkildsen, S. Schauber, T. Busch, and R. Litovsky, "Long-Term Language Development in Children With Early Simultaneous Bilateral Cochlear Implants," *Ear & Hearing*, vol. 41, no. 5, pp. 1294–1305, Sep. 2020. [Online]. Available: https://journals.lww.com/10.1097/AUD.0000000000000851

[8] K. Gfeller, V. Driscoll, M. Kenworthy, and T. Voorst Van, "Music Therapy for Preschool Cochlear Implant Recipients," *Music Therapy Perspectives*, vol. 29, no. 1, pp. 39–49, Jan. 2011. [Online]. Available: https://academic.oup.com/mtp/article-lookup/doi/10.1093/mtp/29.1.39

[9] F.-G. Zeng, Y.-Y. Kong, H. J. Michalewski, and A. Starr, "Perceptual Consequences of Disrupted Auditory Nerve Activity," *Journal of Neurophysiology*, vol. 93, no. 6, pp. 3050–3063, Jun. 2005. [Online]. Available: https://www.physiology.org/doi/10.1152/jn.00985.2004

[10] N. F. A. Shukor, J. Lee, Y. J. Seo, and W. Han, "Efficacy of Music Training in Hearing Aid and Cochlear Implant Users: A Systematic Review and Meta-Analysis," *Clinical and Experimental Otorhinolaryngology*, vol. 14, no. 1, pp. 15–28, Feb. 2021. [Online]. Available: http://e-ceo.org/journal/view.php?doi=10.21053/ceo.2020.00101

[11] C. D. Fuller, J. J. Galvin, B. Maat, D. Başkent, and R. H. Free, "Comparison of Two Music Training Approaches on Music and Speech Perception in Cochlear Implant Users," *Trends in Hearing*, vol. 22, p. 233121651876537, Jan. 2018. [Online]. Available: http://journals.sagepub.com/doi/10.1177/2331216518765379

[12] C. Y. Lo, V. Looi, W. F. Thompson, and C. M. McMahon, "Beyond Audition: Psychosocial Benefits of Music Training for Children With Hearing Loss," *Ear & Hearing*, vol. 43, no. 1, pp. 128–142, Jan. 2022. [Online]. Available: https://journals.lww.com/10.1097/AUD.0000000000001083

[13] M. D. Fletcher, "Can Haptic Stimulation Enhance Music Perception in Hearing-Impaired Listeners?" *Frontiers in Neuroscience*, vol. 15, p. 723877, Aug. 2021. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fnins.2021.723877/full

[14] Gunhyuk Park and Seungmoon Choi, "Perceptual space of amplitude-modulated vibrotactile stimuli," in *2011 IEEE World Haptics Conference*. Istanbul: IEEE, Jun. 2011, pp. 59–64. [Online]. Available: http://ieeexplore.ieee.org/document/5945462/

[15] F. Ganis, M. Vatti, and S. Serafin, "Tickle Tuner - Haptic Smartphone Cover for Cochlear Implant Users' Musical Training," in *Haptic and Audio Interaction Design*, C. Saitis, I. Farkhatdinov, and S. Papetti, Eds. Cham: Springer International Publishing, 2022, vol. 13417, pp. 14–24, series Title: Lecture Notes in Computer Science. [Online]. Available: https://link.springer.com/10.1007/978-3-031-15019-7_2

[16] Z. Duan, C. Gupta, G. Percival, D. Grunberg, and Y. Wang, "SECCIMA: Singing and ear training for children with cochlear implants via a mobile application," in *Proceedings of the 14th Sound and Music Computing Conference, Espoo, Finland*, 2017, pp. 5–8.

[17] J. Jaime, I. Barbancho, C. Urdiales, L. J. Tardón, and A. M. Barbancho, "A new multiformat rhythm game for music tutoring," *Multimedia Tools and Applications*, vol. 75, no. 8, pp. 4349–4362, Apr. 2016. [Online]. Available: http://link.springer.com/10.1007/s11042-015-2478-8

[18] K. Hansen and R. Hiraga, "The Effects of Musical Experience and Hearing Loss on Solving an Audio-Based Gaming Task," *Applied Sciences*, vol. 7, no. 12, p. 1278, Dec. 2017. [Online]. Available: http://www.mdpi.com/2076-3417/7/12/1278

[19] A. Christopoulos and S. Mystakidis, "Gamification in Education," *Encyclopedia*, vol. 3, no. 4, pp. 1223–1243, Oct. 2023. [Online]. Available: https://www.mdpi.com/2673-8392/3/4/89

[20] R. I. Busarello, V. R. Ulbricht, L. M. Fadel, and A. V. De Freitas E Lopes, "Gamification Approaches to Learning and Knowledge Development: A Theorical Review," in *New Advances in Information Systems and Technologies*. Cham: Springer International Publishing, 2016, vol. 444, pp. 1107–1116, series Title: Advances in Intelligent Systems and Computing. [Online]. Available: http://link.springer.com/10.1007/978-3-319-31232-3_105

[21] F. Ganis, A. Gulli, F. Fontana, and S. Serafin, "The Role of Haptics in Training and Games for Hearing-Impaired Individuals: A Systematic Review," *Multimodal Technologies and Interaction*, vol. 8, no. 1, p. 1, Dec. 2023. [Online]. Available: https://www.mdpi.com/2414-4088/8/1/1

[22] G. Cousin, "Case Study Research," *Journal of Geography in Higher Education*, vol. 29, no. 3, pp. 421–427, Nov. 2005. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/03098260500290967

[23] S. Merchel, "Psychophysical comparison of the auditory and tactile perception: a survey," *Journal on Multimodal User Interfaces*, p. 13, 2020.

[24] H. Sharp, Y. Rogers, and J. Preece, *Interaction design : beyond human-computer interaction*, fifth edition ed. Indianapolis, IN: John Wiley & Sons, 2019, publication Title: Interaction design : beyond human-computer interaction.

[25] C. Spinuzzi, "The Methodology of Participatory Design," *Technical communication (Washington)*, vol. 52, no. 2, pp. 163–174, 2005, place: Arlington, VA Publisher: Society for Technical Communication.

[26] H. Lee and D. Müllensiefen, "The Timbre Perception Test (TPT): A new interactive musical assessment tool to measure timbre perception ability," *Attention, Perception, & Psychophysics*, vol. 82, no. 7, pp. 3658–3675, Oct. 2020. [Online]. Available: https://link.springer.com/10.3758/s13414-020-02058-3

[27] Unity Technologies, "Unity," 2021. [Online]. Available: https://unity.com/

[28] Firelight Technologies, “FMOD.” [Online]. Available: https://www.fmod.com/

[29] Cockos, Justin Frankel, “Reaper.” [Online]. Available: https://www.reaper.fm/

[30] F. A. Russo, P. Ammirante, and D. I. Fels, “Vibrotactile discrimination of musical timbre.” *Journal of Experimental Psychology: Human Perception and Performance*, vol. 38, no. 4, pp. 822–826, 2012. [Online]. Available: http://doi.apa.org/getdoi.cfm?doi=10.1037/a0029046

[31] B. Remache-Vinueza, A. Trujillo-León, M. Zapata, F. Sarmiento-Ortiz, and F. Vidal-Verdú, “Audio-Tactile Rendering: A Review on Technology and Methods to Convey Musical Information through the Sense of Touch,” *Sensors*, vol. 21, no. 19, p. 6575, Sep. 2021. [Online]. Available: https://www.mdpi.com/1424-8220/21/19/6575

[32] Daniel J. Velleman and Gregory S. Warrington, “What to Expect in a Game of Memory,” *The American Mathematical Monthly*, vol. 120, no. 9, p. 787, 2013. [Online]. Available: https://www.tandfonline.com/doi/full/10.4169/amer.math.monthly.120.09.787

ASSESSING MUSICAL PREFERENCES OF CHILDREN ON THE AUTISTIC SPECTRUM: IMPLICATIONS FOR THERAPY

Natália SANTOS (up202100737@fe.up.pt) (0009-0006-5596-7077)[1],
Gilberto BERNARDES (gba@fe.up.pt) (0000-0003-3884-2687)[1],
Rita COTTA (rita.alcantara@prof.cesupa.br) (0000-0002-5288-6902)[2],
Nilzabeth COELHO (coord.psicologia@cesupa.br) (0000-0001-5157-0941)[2], and
Alessandra BAGANHA (natasha@cesupa.br) (0000-0002-0010-4825)[2]

[1]**University of Porto, Faculty of Engineering & INESC TEC**, Porto, Portugal
[2]**University Center of the State of Pará-CESUPA**, Belém, Brazil

ABSTRACT

Music-based therapies have been yielding favorable clinical outcomes in children with Autism Spectrum Disorder (ASD). However, there is a lack of guidelines for content selection in music-based interventions. In this context, we propose a methodology for conducting experimental studies on musical preferences in children diagnosed with ASD. It consists of a generative music system with seven manipulable musical parameters where participants are encouraged to create music content according to their preferences. We conducted a preliminary transversal study with 24 children in the state of Pará, Brazil. The results suggest preferences for fast tempo, higher pitch, consonance, high event density, and timbres with smooth attacks. Intriguingly, the results revealed inconsistency in the identified preferences across therapy sessions. The critical need for personalized regulation in music-based interventions for children with ASD highlights the unique nature of individual responses, emphasizing the imperative of tailoring therapeutic approaches accordingly.

1. INTRODUCTION

Autism Spectrum Disorder (ASD) is a neurodevelopmental disorder characterized by poor communication and language skills, restricted and repetitive patterns of behaviour, and persistent deficits in initiating and maintaining reciprocal social interactions. A notable increase in diagnosed cases of ASD has been reported recently [1] – according to the World Health Organization (2023), it is estimated that currently around 1 in 100 children worldwide has autism [2].

In recent years, there has been a remarkable development in research into the disorder, leading to measures for earlier diagnosis and improving these individuals' quality of life, notably from interdisciplinary therapeutic strategies [3]. In this context, available treatments range from pharmacological treatments to behavioral interventions and alternative therapies to promote development and well-being. These interventions are typically carried out by a multidisciplinary team, which includes professionals from the fields of psychology [4], pharmacology [5], speech therapy [6], physiotherapy and psychomotricity [7], occupational therapy [8], pedagogy [9]. Commonly applied methods for teaching and improving language, psychomotor regulation, communication, socialization and adaptive behaviors include Applied Behavior Analysis [4, 10, 11], Picture Exchange Communication System [12] , TEACCH [4, 12], Early Start Denver Model [10], Relationships, Individual Differences and Development Model [4, 10], and Social Skills Training [10].

Additionally, professionals are increasingly opting for sessions using alternative therapies, which are health practices that are not part of tradition or conventional medicine. One of the alternative therapies that has shown promising results uses music as a form of therapeutic intervention [13]. Music therapy is an intervention with music in a clinical treatment context, used through a therapist-patient relationship to promote individualized goals [14]. This intervention employs a wide range of methods and techniques, which are characterized by systematic procedures that specify the particular way in which the technique is to be performed [15]. In the context of autism, common music therapy methods include Educational Music Therapy [16], Improvisational Music Therapy [15, 16], Singing/Listening to Songs [15], Audio Murotal Intervention [17], Therapeutic Listening [17] and Auditory Integration Training [17].

Although the use of music therapy has shown significant improvements in the areas of communication, social skills, attention, emotional expression, and sensorimotor regulation, there is still a lack of in-depth understanding about the most appropriate sounds for these interventions [18]. In this regard, it is becoming increasingly urgent to understand the sound and musical preferences of individuals with this disorder, to enable the development of individualized musical approaches and sound stimuli, capable of meeting their demands and specificities [18, 19].

This experimental study delves into how to prepare sound stimuli and/or music to be effectively incorporated into ASD therapies, and it was conducted as part of a larger project at the University Center of the State of Pará (CE-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

SUPA) that aimed to develop a mobile application for children with ASD, focusing on their emotional and psychomotor regulation. At the same time, this study set out to investigate: 1) what the preferences are within the various musical parameters, as well as 2) whether there are any correlations between them, and also 3) whether this preference is constant or changes between therapy sessions, for children with ASD, aged between 2 and 6, of both sexes and with support levels 1 or 2, according to the DSM-5-TR [20].

We designed a methodology based on a generative computational musical engine, running on a touch screen mobile handheld device (MHD), where participants were asked to manipulate generative musical parameters.

The remainder of this paper is structured as follows. Section 2 presents related work on musical preferences in ASD. Section 3 outlines the methodology adopted in an experimental study, including participants, materials, procedures, and data analysis. Section 4 presents the results of our experiment, followed by a discussion of their implications within the context of existing literature in Section 5. Finally, Section 6 provides the main conclusions of our study and future research directions.

2. RELATED WORK ON MUSICAL PREFERENCES IN ASD

A growth literature has been reporting music interventions for individuals within the ASD. They range from reviews of the various types of music interventions [15, 16, 21], to studies comparing specific types of music interventions [16, 17], and also studies exploring music interventions for particular aspects of ASD [17, 22]. A common thread among these interventions for individuals with ASD is the need for curated music contents. Nonetheless, there remains a lack of clear guidelines for selecting appropriate musical content, particularly when initiating therapy with new patients. The sparse research exploring musical preferences among individuals with ASD examines a few parameters, concentrating mostly on higher-level genre preferences compared to lower-level structural elements such as consonance, melody, rhythm, and instrumentation. This section synthesizes findings across these parameters to provide insights into the nuanced musical preferences of individuals with ASD.

Genre preferences serve as a foundational aspect of musical preference studies among individuals with ASD. Research by Bhatara et al. [23] indicates a similar interest in music among adolescents with and without ASD, with a notable preference for classical music among those with the disorder. Levasseur et al. [19] found that rock and pop-rock were favored genres among children with ASD, while Sravanti et al. [24] uncovered a liking for culturally significant music from an early age.

Moving to more formal parameters, investigations into consonance versus dissonance perception reveal intriguing findings. Boso et al. [25] demonstrated a preference for consonant music over dissonant music among both individuals with ASD and control groups, suggesting the therapeutic potential of consonant music in communication and interaction.

Further exploration into musical structure includes preferences for melody versus rhythm. Sravanti et al. [24] reported a preference for rhythm over melody among children with ASD, with some expressing enjoyment of both aspects of music.

Additionally, preferences for instrumental music versus music with vocals have been studied. Sravanti et al. [24] found a clear preference for music with vocals over instrumental music among children with ASD, highlighting the importance of considering vocal elements in therapeutic interventions.

Finally, research into sound parameters provides information on the sensory experiences of individuals with ASD. Cibrian et al. [18] found a preference for melodic sounds, particularly those produced by the cello, as well as a preference for low notes over high notes, suggesting potential applications for the regulation of attention and emotion in therapeutic contexts.

Overall, these studies underscore the complexity of musical preferences among individuals with ASD, encompassing both higher-level genre preferences and more formal structural parameters. While providing valuable insights, further research is needed to fully understand the intricacies of musical preferences in ASD and to inform tailored interventions in therapeutic settings.

3. METHODS

We conducted an experimental study to unveil musical preferences of children with ASD regarding the parameterization of an instrumental only musical structure. We adopted a differentiated approach, contrary to existing studies that primarily employ third-person observations and perceptual surveys reflecting on the musical experience. These may fail to provide insightful or valid results due to the subjective nature of the task and heavy reliance on working memory, as individuals with ASD may have deficits in short-term (working) memory, impacting their ability to, for example, follow multi-step instructions and maintain focus on tasks requiring mental manipulation of information [26].

In this context, we propose a novel methodology for inferring musical preferences in ASD individuals by assessing trends in the parameterization of a generative music engine intuitively controlled on a MHD. The playful nature of the generative application aims to attract ASD children, the target population of our study, while allowing inferring preferences of the following seven musical parameters: tempo, pitch height, consonance, (vertical) note density, timbre, percussivity, and loudness.

3.1 Participants

The study was carried out from January 2023 to May 2023, and took place at the Physiotherapy School Clinic of CESUPA (Pará State University Center). We recruited 24 children (8% female), aged between 3 and 6 years old (mean 5 ± 3.19), who had a proven diagnosis of ASD, with support level 1 or 2 (29%), according to the DSM-5-TR clas-

sification [20] and with psycho-motor repercussions. The participants were voluntarily enrolled in the study, and all the parents consented to the study on behalf of their children.

3.2 Materials

Figure 1 shows the generative music application created for the study. The basic code for the application was created in Pure Data [27] and was based on the harmonic sequence generation algorithms Conchord [28] and D'accord [29]. The algorithms were expanded to allow for a wider range of musical layers (e.g. melody and rhythm section), as well as the parameterization of musical elements.

The musical content produced by the MHD application bears resemblance to the pop-rock and electronic genres. The timbres encompass a range of sonorities reminiscent of electric guitars, electronic pianos, marimbas, among others, while the percussion is derived from samples reminiscent of rap-rock style. To facilitate a deeper understanding of the layers within the musical content generated by the application used in the experiment, we provide musical examples online at `10.6084/m9.figshare.25377460`.

Generated music results from the parameterization of seven musical parameters:

1. **Tempo**, in beats per minute (bpm) in the continuous 40 to 140 range;

2. **Pitch** in semitones, defining a transposition interval for the harmonic and melodic content, ranging [-19,19] semitones;

3. **Consonance (Cons)** of the generated harmonic content as measured by the tonal consonance indicator in the Tonal Interval Space [30] in the continuous [0.4, 0.6] range;

4. **Density (Dens)**, defining the number of notes per unit of time in harmonic and melodic layers. It adopts a 5-point scale, ranging from zero to four;

5. **Timbre** qualities of the sound attack envelope, ranging from static smooth sound attacks transients to percussive sounds or sounds with a fast attack transient on the harmonic and melodic layers. It adopts a 6-point scale, ranging from zero to five;

6. **Percussivity (Perc)** defines the number of percussive elements adopted in the rhythmic layer by swapping from no percussion to 8 percussive loops with increasing degree of event density;

7. **Loudness (Loud)** controls the sound volume of all the generated layers, in the continuous [0,1] range.

Figure 1 displays the main interface of the MHD application. The design process commenced by selecting white as the base color due to its neutral nature. Knobs, circular buttons designed for adjusting musical parameters, were

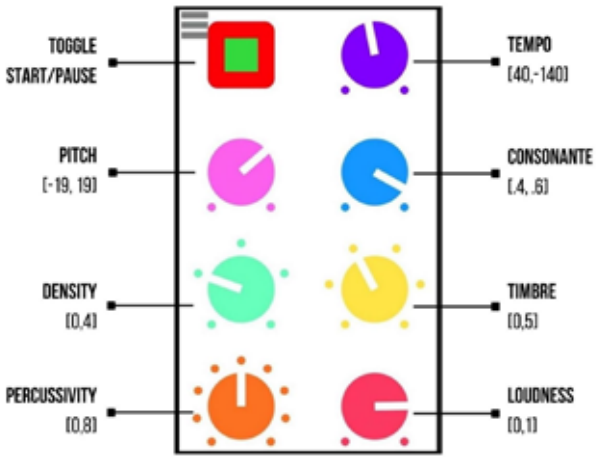


Figure 1. Interface of the mobile generative music application used in the experiment to infer the musical preferences of children with ASD.

implemented for their intuitive usability, particularly beneficial for children. Each knob was assigned a distinct color, aiding children in recognizing that each control modifies a specific musical parameter. Additionally, color selection took into account findings from prior studies suggesting colors with positive effects on children with ASD [31–33]. Furthermore, a Toggle button was incorporated to initiate or halt sound playback, featuring a red hue when paused and green when activated.

3.3 Procedures

The experience began after approval from Cesupa's Ethics Committee, and it was part of a broader project conducted the University Center of the State of Pará (CESUPA), whose objective was the development of a mobile application aimed at children with ASD, focused on their emotional and psychomotor regulation. Participants had one or two intervention sessions, with those receiving two sessions having them on different days, spaced one to two weeks apart. During the initial project phase, lasting about 1 minute per session, children were briefed on the functionality of the MHD application, including its various control parameters. They were asked to experiment only with manipulating them, without concentrating on the musical result. Subsequently, there were two phases in which the children were subjected to physiotherapy and psychology interventions. Each one lasted 14 minutes and aimed to stimulate and work on particular aspects of the child's development.

The experience of this study took place in the last phase of this larger project. The experiment was conducted by the therapist, so that there was a common link to the ASD children. After the physiotherapy and psychology interventions, the participants were asked to create music content according to their taste by manipulating the musical parameter controls of the MHD application. This phase lasted about two minutes, during which the participants experimented with the various musical parameters in the MHD application, until they produced a final music con-

tent that was based on their particular taste. This task was performed completely independently and autonomously by the participants, unless they revealed difficulties in the use of the controls, in which case they were assisted by the therapist in manipulating the parameter controls, without interfering in the final choice of the participants. Once a final parameter setting was achieved, the therapist wrote down in a specific table the values chosen by the child for each of the musical parameters, giving rise to the data for later analysis.

3.4 Data Analysis

We adopt a threefold analysis to the data to assess: 1) preferences for specific musical parameters for the participants sample; 2) infer dependencies across musical parameters; and 3) verify consistency in preferences across the two sessions. While the parameters preferences can lead to guidelines to adopt musical parameters spaces in musical stimuli for children with ASD, the second analysis infers the coupling of parameters, such as in fast tempos, typically higher pitch content is adopted. Finally, the consistency of preferences across session should inform the reliability of data across time per participant.

To infer preferences per musical parameters, we computed descriptive statistics, determining the mean, std% and skewness per parameter. The mean indicated the general trend, while the standard deviation assessed consistency between participants. Skewness identified asymmetry in the distribution and was useful in detecting tendencies in the data. Namely, parameters exhibiting low degrees of standard deviation denote greater agreements in the mean value across participants.

Skewness values closer to either negative or positive one, lead to parameters that have a certain tendency across categorized data, e.g., faster or slower tempo, and higher or lower density. Conversely, skewness value closer to zero denote symmetrical distributions. To further interpret the skewness, we computed histograms on data aggregated by fewer categories, such as a threefold categorization of slow, moderate fast tempos, to offer an intuitive and categorized distributions per parameter.

To explore possible relationships between musical parameters, the Pearson's correlation coefficient r was computed between all parameter pairs. This coefficient, in the $[-1, 1]$ range, can offer insights into the dependencies between parameters. Pearson correlation results near positive or negative unity indicate linear or inverse relationships between the parameters data, thus denoting some dependency across parameters. The closer to zero the less of a relationship exist between parameter pairs. Results can inform in the design of musical stimuli for ASD therapies.

To assess the consistency in preferences across different sessions per participant, Pearson's correlation coefficient r and its significance level p as well as the coefficient of determination R^2 were applied. The coefficient of determination, ranging from zero to one, quantifies the degree of relationship between the dependent and independent variables, allowing us to determine whether preferences are consistent over time. Values close to one indicate a strong relationship, while values close to zeo suggest a lack of significance. Prior to the correlation computation, we applied the L_1 norm to the data per parameter, scaling all values to the same range, thus avoiding erroneous linear correlations from outlier data.

4. RESULTS

The data collected from the 24 participating children underwent pre-processing to filter out unreliable interactions with the MHD application, based on feedback provided by the therapists. Initially, we excluded data from children who did not engage with MHD application. Subsequently, following a brief analysis of the remaining data, we further eliminated instances exhibiting a consistent visual pattern, such as consistently reaching maximum or minimum values. This exclusion criterion was based on the observation that these patterns tended to disregard the musical feedback, focusing solely on visual cues. Finally, data from 18 participants were considered for analysis. The first session had 15 participants and the second session had 12 participants. Nine participants attended both sessions.

4.1 Musical Parameters Preferences

Table 1 presents the descriptive statistics for the filtered data. Participants are represented by the symbol P, followed by a unique numerical identifier. For instance, participant one is denoted as P.ID01. The table is divided into two sections, the above results are from session 1 and the bottom results from session 2.

Based on the analysis that parameters exhibiting lower standard deviation values across all participants indicate more representative median parameter values, we can highlight greater alignment in the participants preferences in terms of consonance (13.5 STD%), tempo (33.6 STD%) and intensity (42.1 STD%), resulting in a preference for consonance sounds, faster tempos around 102 bpm and medium loudness. On the other hand, the musical parameters pitch and timbre showed greater divergence between the participants' choices, with higher percentage standard deviations (389.5% and 82.9%, respectively).

The asymmetry analysis indicated a consistent trend for most of the musical parameters, except for percussivity, which showed no trend. The categorical histogram analysis in Figure 2 highlights trends in preference for specific categories of values per musical parameter. The analysis corroborates the data in the Table 1, showing that parameters such as consonance, tempo and intensity were consistent among the participants, while pitch and timbre showed greater variation in preferences. The histograms confirm a preference for high tempos, high consonance sounds, medium intensity, high density and more sustained timbres. The pitch histogram showed no clear trend, and as with the skewness values for percussivity, the data indicated a general appreciation of percussion, but no specific preference for types of percussion was observed, which is why we chose not to include the graph.

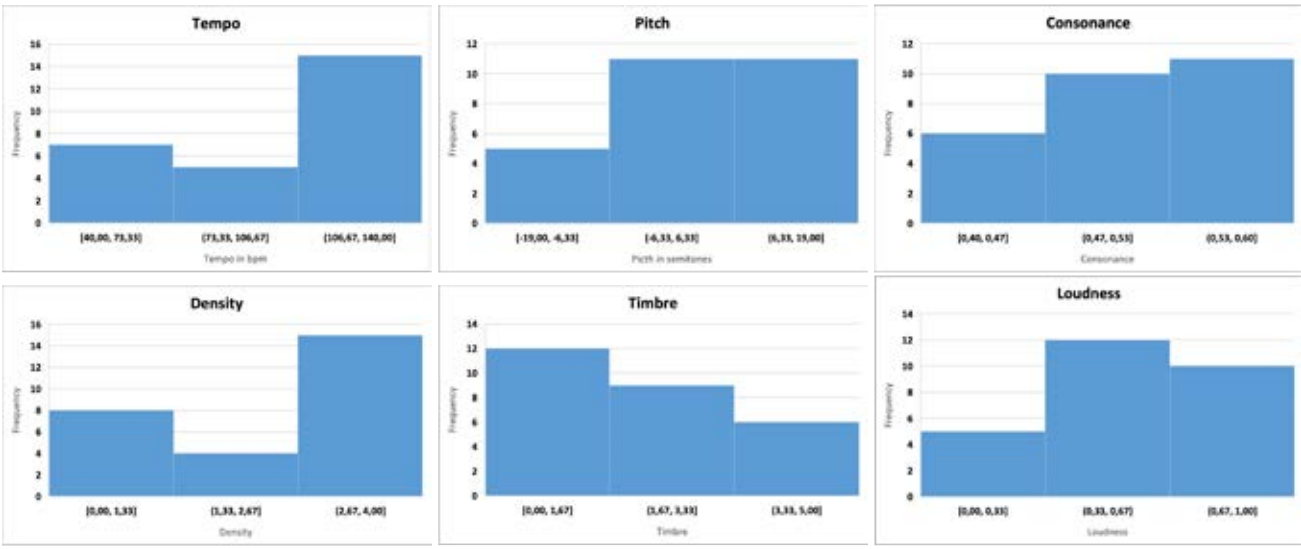


Figure 2. Histograms of musical parameter preferences from the ASD population under study to identify trends in categorical data.

ID	Tempo [40,-140]	Pitch [-19,19]	Cons [.4,.6]	Dens [0-4]	Timbre [0,5]	Perc [0,8]	Loud [0,1]
P.ID01	40	14	.56	3	5	0	.83
P.ID02	140	18	.4	1	0	0	.97
P.ID03	75	-4	.6	0	2	1	0
P.ID04	47	19	.6	4	3	0	.66
P.ID05	107	-19	.59	0	1	4	.45
P.ID06	140	-1	.49	2	2	4	.51
P.ID07	137	18	.53	3	0	2	.56
P.ID08	88	0	.4	4	1	2	.65
P.ID09	107	6	.53	3	3	5	.58
P.ID10	40	19	.4	4	0	5	1
P.ID11	76	-2	.48	2	3	6	1
P.ID12	107	6	.53	3	3	5	.58
P.ID13	136	17	.6	4	5	8	0
P.ID14	140	17	.6	4	5	8	1
P.ID15	47	-19	.4	0	1	8	.48
P.ID01	67	1	.49	0	1	8	.83
P.ID02	128	4	.5	0	0	8	.68
P.ID03	133	19	.47	1	1	1	.72
P.ID05	140	0	.59	3	2	3	.32
P.ID07	137	16	.4	3	5	4	.49
P.ID08	92	0	.52	2	3	4	.5
P.ID09	46	-12	.59	0	3	8	.47
P.ID11	115	-11	.6	4	5	2	.96
P.ID15	88	0	.4	4	1	2	.65
P.ID16	118	10	.56	3	4	6	.73
P.ID17	113	-9	.6	2	0	4	.16
P.ID18	44	14	.47	4	1	7	.28
Mean	101.75	2,67	.52	2.17	2.17	4.75	.57
STD%	33.6	389.5	13.5	73.3	82.9	53.9	42.1
Skew	-.39	-.41	-.39	-.47	.36	-.01	-.43

Table 1. Data resulting from our experiment compiling the parameters of each participant in two sessions from manipulating a generative music application. An horizontal line splits the data between session one (above) and session two.

4.2 Parameter dependency

To check for possible linear relationships between the parameters, we used Pearson's Correlation Coefficient r, which measures the degree of correlation between every parameter pair variables whose results are shown in Table 2. There are no linear relationships (positive or negative) between any of the parameter pairs. However, two values denote a relative relationship with a high positive trend, namely between the variables density versus pitch ($r = .5$), and timbre versus consonance ($r = .42$).

	Pitch	Cons	Dens	Timbre	Perc	Loud
Tempo	.18	.12	.06	.08	-.07	-.06
Pitch		-.16	.50	.13	-.20	.24
Cons			0	.42	.03	-.32
Dens				.38	-.18	.14
Timbre					.08	.06
Perc						-.09

Table 2. Pearson's correlation coefficient r across all parameter pairs assessed in the experiment for inferring linear relationships between parameters.

To further examine the parameter pairs exhibiting linear dependencies between parameters, we present scatter plots accompanied by their linear regressions for density versus pitch in Figure 3 and timbre versus consonance in Figure 4. The positive linear correlation observed between density and pitch indicates that as music density rises, pitch tends to increase as well. Consequently, denser music tends to favor higher-pitched structures. Likewise, the positive linear relationship between timbre and consonance suggests that as sharper attacks are employed in the timbre dimension, the more consonant the structures are.

4.3 Participant's consistency across sessions

Finally, to check whether each participant's preferences for specific musical parameters remained constant in both

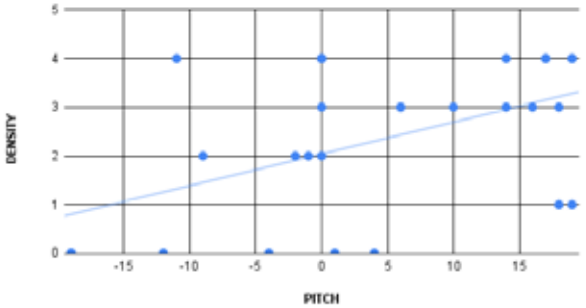


Figure 3. Scatter plot representing the relationship between the density and pitch parameters.

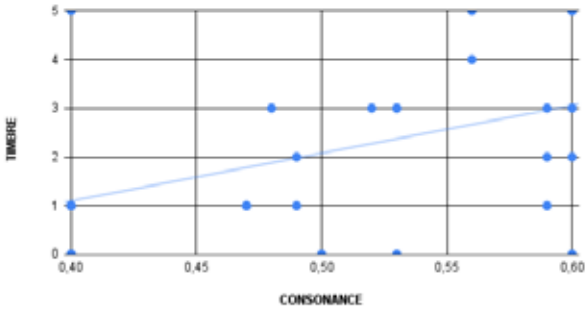


Figure 4. Scatter plot representing the relationship between the timbre and consonance parameters.

sessions, the Pearson's Correlation Coefficient and Coefficient of Determination were computed across the data collected per participants in both sessions. Only data from participants who took part in two sessions was considered. Of the 18 participants, nine were included in this analysis, and the results are shown in Table 3.

ID	r	p	R^2
P.ID01	.09	.11	.01
P.ID02	.75	.77	.56
P.ID03	-.04	.65	0
P.ID05	.75	.05	.57
P.ID07	-.05	.83	0
P.ID08	-.85	.60	.72
P.ID09	-.63	.19	.40
P.ID11	.28	.09	.08
P.ID15	-.34	.11	.11

Table 3. Pearson's correlation coefficient and Coefficient of Determination of the values chosen by the participants submitted to two sessions, to infer consistency between sessions.

Globally, results show poor agreement between the sessions with non-correlated data in r and large variability in R^2. Only one participant in Table 3, P.ID05, shows relatively consistent results between the two sessions, i.e., the parameters choice between the two session are linearly correlated.

5. DISCUSSION

This study explored musical parameter preferences in children with ASD and investigate potential linear relationships between these parameters, as well as the consistency of preferences across different sessions for the same participant. Our aim was to identify trends that could be adopted as guidelines for designing musical stimuli for ASD therapy.

In our results, it was noted that children with ASD exhibited a preference for higher pitch, diverging from Cibrian et al. [18] findings, which suggested higher attention to lower notes. The disparity can be attributed to variations in study objectives, methodologies, and session duration. Concerning consonance, children with ASD favored more consonant music, aligning with Boso et al. [25] previous findings. No specific percussivity preference was identified, but a propensity for using and preferring percussive elements emerged, consistent with Sravanti et al. [24] research.

High-density music was preferred, resonating with studies linking ASD preferences to energetic genres like rock and pop-rock [19, 23, 24]. Sustained timbres were favored over percussive ones, in line with Cibrian et al. [18] observation of heightened concentration and positive emotions with sustained sounds. Participants showed a stronger preference for faster tempos, although some preference for slower tempos was noted compared to medium tempos. Medium loudness values were preferred, potentially linked to the participants' ability to regulate music volume.

Comparing data from two sessions revealed varied individual preferences, emphasizing the complexity of sound preferences in children with ASD. The study underscores the importance of incorporating individual musical preferences into the design of interventions, offering insights for more effective therapeutic strategies. Recognizing the specific preferences of each child holds the potential to enhance the effectiveness of these strategies, create a more fitting and captivating musical environment, and contribute valuable information to the field. Despite acknowledging limitations such as sample size and population specifics, the findings highlight the need for ongoing exploration of preferences within the diverse ASD population. Additionally, a potential limitation is that the protocol could indicate preferences in a creative and generative situation, which may not necessarily correspond to the subject's musical preferences when listening to music without actively participating. Therefore, further research is needed to validate these findings in more passive listening contexts and ensure the robustness of personalized interventions. There is an imperative need to persist in expanding knowledge in this domain, considering the diverse range of preferences within the ASD population.

6. CONCLUSIONS

This study explored the musical preferences of children on the ASD and investigated the impact of different mu-

sical parameters on their therapeutic response. The results revealed that children with ASD showed specific musical preferences concerning various sound elements. The participants showed a preference for music with faster tempos, higher pitch, more consonant music, high density, more sustained timbres and percussive elements. Positive linear relationships were observed between music density and pitch, as well as between timbre and consonance. Finally, the results also showed that the consistency of musical preferences in the experimental method sessions varied greatly between sessions.

Our findings can contribute to clinical practice, offering insights for professionals working in the field of music therapy with ASD children by recognizing and understanding the need for individualized, targeted musical content for each patient. Despite the poor agreement across sessions per participant in our results, the above parameter guidelines can provide a solid starting point for content creation. In this context, arising from these experiment-derived patterns, a series of musical pieces were produced for therapy applications, which are accessible online at `10.6084/m9.figshare.25377493`. Furthermore, the experimental methodology adopted can foster creative direction for interacting with ASD children, providing a high degree of control over the generated musical structure and preferences. Longitudinal studies examining musical parameter preferences per ASD child ought to inform us about better practices in this domain.

In future work, we aim to explore the relationship between musical preferences and behavioral responses, investigating the impact of sensory sensitivity, comparing preferences across different ASD profiles and assessing the effectiveness of personalized music therapy. Additionally, we want to apply the protocol to children without ASD to infer differences to children outside of the ASD. Furthermore, the use of innovative technologies, such as applications, can be explored for evaluating and incorporating musical preferences into interventions.

7. ETHICAL STATEMENT

The research was approved by the Ethics Committee of the University Center of the State of Pará, CAAE 65135122.0.0000.5169.

8. REFERENCES

[1] M. F. do Nascimento Araújo, I. K. dos Santos Barbosa, A. T. P. de Holanda, C. S. de Moura, J. B. de Barros Santos, V. S. da Silva, I. T. dos Santos, J. K. G. Campelo, A. G. da Silva, and É. M. do Nascimento Silva, "Autismo, níveis e suas limitações: uma revisão integrativa da literatura," *PhD Scientific Review*, vol. 2, no. 05, pp. 8–20, 2022.

[2] World Health Organization, "Autism spectrum disorders," https://www.who.int/news-room/fact-sheets/detail/autism-spectrum-disorders, November 15 2023, who.int; World Health Organization: WHO.

[3] S. B. Sulkes, "Golisano children's hospital at strong, university of rochester school of medicine and dentistry," *Transtornos do espectro autista*, 2022.

[4] S. S. de Abreu Almeida, B. P. G. S. Mazete, A. R. Brito, and M. M. Vasconcelos, "Transtorno do espectro autista," *Residência Pediátrica*, 2018.

[5] H. H. P. Almeida, J. P. de Lima, and K. B. N. T. Barros, "Cuidado farmacêutico às crianças com transtorno do espectro autista (tea): contribuições e desafios," *Encontro de Extensão, Docência e Iniciação Científica (EEDIC)*, vol. 5, no. 1, 2019.

[6] C. A. d. l. H. Amato, T. H. F. Santos, M. R. P. Barbosa, and F. D. M. Fernandes, "Estudo longitudinal da terapia de linguagem de 142 crianças e adolescentes com distúrbios do espectro do autismo," in *CoDAS*, vol. 25. SciELO Brasil, 2013, pp. 388–390.

[7] A. A. d. Oliveira *et al.*, "Intervenções fisioterapêuticas e de terapia ocupacional para participação social de crianças e adolescentes com transtorno do espectro autista: revisão integrativa," 2022.

[8] B. A. Mapurunga, A. L. R. Mendes, V. B. Silveira, R. F. de Oliveira Correia, and A. F. M. de Carvalho, "A atuação do terapeuta ocupacional na reabilitação de pessoas com autismo," *Revista de Casos e Consultoria*, vol. 12, no. 1, pp. e26 291–e26 291, 2021.

[9] M. I. B. Monteiro and J. M. B. Bragin, "Práticas pedagógicas com autistas: ampliando possibilidades," *Journal of Research in Special Educational Needs*, vol. 16, pp. 884–888, 2016.

[10] D. A. Zachor, P. Curatolo *et al.*, "Recommendations for early diagnosis and intervention in autism spectrum disorders: An italian–israeli consensus conference," *European journal of paediatric neurology*, vol. 18, no. 2, pp. 107–118, 2014.

[11] E. Baker and S. S. Jeste, "Diagnosis and management of autism spectrum disorder in the era of genomics: rare disorders can pave the way for targeted treatments," *Pediatric Clinics*, vol. 62, no. 3, pp. 607–618, 2015.

[12] C. A. Bosa, "Autismo: intervenções psicoeducacionais," *Brazilian Journal of Psychiatry*, vol. 28, pp. s47–s53, 2006.

[13] C. C. R. d. Silva *et al.*, "Música: um auxílio no desenvolvimento e aprendizagem de crianças com a perturbação do espectro do autismo," Master's thesis, 2012.

[14] K. Bruscia, "Definindo musicoterapia. barcelona," 1998.

[15] A. V. Marquez-Garcia, J. Magnuson, J. Morris, G. Iarocci, S. Doesburg, and S. Moreno, "Music therapy in autism spectrum disorder: A systematic review," *Review Journal of Autism and Developmental Disorders*, pp. 1–17, 2021.

[16] H. Mayer-Benarous, X. Benarous, F. Vonthron, and D. Cohen, "Music therapy for children with autistic spectrum disorder and/or other neurodevelopmental disorders: a systematic review," *Frontiers in psychiatry*, vol. 12, p. 435, 2021.

[17] F. A. Shahrudin, A. A. A. Dzulkarnain, A. M. Hanafi, F. N. Jamal, N. A. Basri, S. Na'im Sidek, H. M. Yusof, and M. Khalid, "Music and sound-based intervention in autism spectrum disorder: A scoping review," *Psychiatry Investigation*, vol. 19, no. 8, p. 626, 2022.

[18] F. L. Cibrian, J. Mercado, L. Escobedo, and M. Tentori, "A step towards identifying the sound preferences of children with autism," in *Proceedings of the 12th EAI international conference on pervasive computing technologies for healthcare*, 2018, pp. 158–167.

[19] A. Levasseur, K. Jamey, N. Foster, K. Hyde, and S. Dalla Bella, "Musical genre preferences in autism spectrum disorder," in *NeuroMusic Annual Conference*, 2020.

[20] A. P. Association, *Diagnostic and Statistical Manual of Mental Disorders: DSM-5-TR*. American Psychiatric Association Publishing, 2022.

[21] C. P. De Barros Freitas and K. Figueira, "Efeito da musicoterapia nas perturbações do espectro do autismo: uma revisão baseada na evidência," 2018.

[22] T. Wigram and C. Gold, "Music therapy in the assessment and treatment of autistic spectrum disorder: clinical application and research evidence," *Child: care, health and development*, vol. 32, no. 5, pp. 535–542, 2006.

[23] A. Bhatara, E.-M. Quintin, E. Fombonne, and D. J. Levitin, "Early sensitivity to sound and musical preferences and enjoyment in adolescents with autism spectrum disorders." *Psychomusicology: Music, Mind, and Brain*, vol. 23, no. 2, p. 100, 2013.

[24] L. Sravanti, J. V. S. Kommu, S. Suswaram, and A. S. Yadav, "Musical preferences of indian children with autism spectrum disorder and acceptability of music therapy by their families: An exploratory study," *Industrial Psychiatry Journal*, vol. 32, no. 1, p. 176, 2023.

[25] M. Boso, M. Comelli, T. Vecchi, F. Barale, and P. Politi, "Exploring musical taste in severely autistic subjects: preliminary data," *Annals of the New York Academy of Sciences*, vol. 1169, no. 1, pp. 332–335, 2009.

[26] P. Desaunay, A. R. Briant, D. M. Bowler, M. Ring, P. Gérardin, J.-M. Baleyte, F. Guénolé, F. Eustache, J.-J. Parienti, and B. Guillery-Girard, "Memory in autism spectrum disorder: A meta-analysis of experimental studies." *Psychological Bulletin*, vol. 146, no. 5, p. 377, 2020.

[27] M. S. Puckette *et al.*, "Pure data," in *ICMC*, 1997.

[28] G. Bernardes, D. Cocharro, C. Guedes, and M. E. P. Davies, "Conchord: An application for generating musical harmony by navigating in the tonal interval space," in *Music, Mind, and Embodiment*, R. Kronland-Martinet, M. Aramaki, and S. Ystad, Eds. Cham: Springer International Publishing, 2016, pp. 243–260.

[29] G. Bernardes, D. Cocharro, C. Guedes, and M. E. Davies, "Harmony generation driven by a perceptually motivated tonal interval space," *Computers in Entertainment (CIE)*, vol. 14, no. 2, pp. 1–21, 2016.

[30] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[31] M. Grandgeorge and N. Masataka, "Atypical color preference in children with autism spectrum disorder," *Frontiers in psychology*, vol. 7, p. 1976, 2016.

[32] L. C. Moreno, "A influência das cores no desenvolvimento de crianças autistas," *Revista Científica Arqui-Engenharia e Análise e Desenvolvimento de Sistemas*, vol. 1, no. 1, pp. 11–23, 2018.

[33] R. S. Pietra-re and D. de Interiores-Ambientação, "A influência das cores e materiais para as crianças autistas, no âmbito escolar," 2018.

AN ACCESSIBLE SOFTWARE INTERFACE FOR COLLABORATIVE MUSIC PERFORMANCE

Vanessa FASCHI (vanessa.faschi@unimi.it) (0000-0002-9815-1127) [1],
Luca Andrea LUDOVICO (luca.ludovico@unimi.it) (0000-0002-8251-2231) [1],
Federico AVANZINI (federico.avanzini@unimi.it) (0000-0002-1257-5878) [1],
Emanuele PARRAVICINI (emanuele@audiomodeling.com) [2], and
Manuele MAESTRI (manuelemaestri@gmail.com) [3]

[1] *Laboratory of Music Informatics (LIM)*, *Department of Computer Science*, **University of Milan**, via G. Celoria 18, Milan, 20133 Italy
[2] **Audio Modeling**, Via Bernardino Luini 65, Meda, 20821 Italy
[3] **Musica Senza Confini**, www.musicasenzaconfini.com, Italy

ABSTRACT

This paper describes an observational study focusing on an accessible musical instrument software prototype. This interface was designed, on the one hand, with no requirements concerning the level of musical knowledge and ability of users, and, on the other side, to overcome possible physical and cognitive impairments. Specifically, the software prototype implements a MIDI controller that supports a wide range of common input devices (e.g., joysticks, game controllers, keyboards, eye-tracking systems, etc.) to set musical parameters. The prototype was developed in *JUCE*, an open-source and multi-platform framework. The results of the observational study were expected to unveil how robust and ready-to-use the system was perceived by various categories of users. To this end, we involved both impaired and non-impaired users, moreover presenting different degrees of musical knowledge. Usability tests showed an average level of flexibility in supporting and mapping a wide range of heterogeneous controllers. The prototype was able to obtain preparatory feedback to broaden its development in the field of musical parameter control.

1. INTRODUCTION

Accessibility is defined as the ability of an individual to enter, navigate, understand, and use the characteristics of an environment, including physical, digital, and conceptual environments [1]. Within the broad scope of accessibility, computer applications impose highly specific demands [2]. According to [3], the majority of software tools operate under the assumption that users possess the capability to effortlessly execute tasks such as reading and responding to on-screen text and images, typing on a conventional keyboard, selecting text, images, and other data using a mouse, and responding to auditory cues. On the contrary, people with special needs may encounter challenges in performing one or more of these actions, thereby hindering their ability to utilize even widely adopted computer applications, either partially or entirely. Accessibility is a theme that also recurs in the development and implementation of musical instruments and interfaces. Musical activities, be it simply listening, composing, or performing a piece, are, in fact, known to produce benefits, since music is a source of pleasure [4] and a universal means of communication.

The term *usability* is frequently interwoven with discussions on accessibility, representing concepts such as fit to use, functionality, operability, serviceability, validity, and operationality. While usability primarily pertains to the satisfaction of functional requisites, it diverges from the concept of accessibility. However, certain scholars employ both terms interchangeably, positing that they are commonly delineated through observed task performance and collectively embody the notion of person-environment congruence [5]. A comprehensive exploration of accessibility, usability, and universal design is provided by Iwarsson and Ståhl [6].

Another important term underlying our research is *inclusivity*, understood to be a fair and equal treatment to all groups of individuals, without distinction of different races, ethnicities, religions, abilities, genders, and sexual orientations. [1] Accessibility, inclusivity, and usability are the pillars of our work.

Information technology can be an important facilitator of social inclusion for people with disabilities [7]. Thanks to digital technologies, users can overcome their impairments and can be put in the condition of creating and playing music using their physical and/or cognitive abilities. Unfortunately, digital technologies for music expression are limited in number and often expensive, due to their restricted market, the request for high specialization, and the expected return on investment from the industry.

From these considerations arises our mid-term design goal of breaking down the barriers to the musical expressiveness of people with disabilities. The project involves the design, implementation, and release of a hardware

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] https://dictionary.cambridge.org/

and software platform, called *Inclusive MIDI Controller*, which acts as a multi-layer ecosystem that includes an operating system, a vocal assistants to support and profile users, MIDI communication capabilities, data-collection APIs, and AI-based features. The long-term goal is to create a solution that can scale up to sustain the abilities and overcome the disabilities of users, allowing them to employ the interface devices to which they are already accustomed in their daily activities. For this purpose, we also plan to exploit the capabilities of MIDI 2.0, and, in particular, the discovery feature that allows devices to communicate and exchange information each other. In fact, MIDI 2.0 devices can automatically configure themselves to work with each other, reducing the need for manual setup and configuration. The focus of this paper is to test a prototype implementation of *Inclusive MIDI Controller* with some representative users as it regards accessibility and usability.

This work arises from the collaboration of three institutions. *Musica Senza Confini* [2] is a musical project that addresses children and adults with psycho-physical disabilities and deals with inclusive music production. Based on the experience with people in condition of quadriplegia, amyotrophic lateral sclerosis, and different types of cerebral palsy, *Musica Senza Confini* provided the concept and designed the software prototype of the accessible musical instrument described in this paper. *Audio Modeling* [3] is a company specialized in expressive digital emulation of acoustic instruments. One of their products is a multi-platform application for live performances that supports the connection of hardware and software plugin instruments and the integration of musical scores. Audio Modeling is developing and planning to release the platform described in this study. The *Laboratory of Music Informatics* (LIM) at the University of Milan [4] provides expertise in sound and music computing. In the framework of this project, the lab's activities focused on software development, usability tests, accessibility research, and MIDI communication among devices.

The remainder of the paper is structured as follows. Section 2 presents an analysis of the scientific literature on the subject, surveying the state of the art and identifying current trends and innovations. Section 3 delves into *Inclusive MIDI Controller*, delineating the design of the interface and the technologies employed. Section 4 focuses on an observational study conducted on three subjects, addressing the research question, research protocol, testing methodology, and results obtained from empirical observations. Lastly, in Section 5, we synthesize our discoveries and offer conclusive insights drawn from our exploration.

2. STATE OF THE ART

In this section, we will analyze some relevant examples of digital technologies and tools to overcome impairments in the music experience and expression.

Accessible digital musical instruments (ADMIs), intended as accessible musical control interfaces used in electronic music, inclusive music practice, and music therapy settings [8], are becoming more and more investigated in the scientific literature [9]. One of the pioneering examples was a "biocontroller" designed in 1990 to increase the performance of musical instruments and to regain the pleasure of making music in people with motor impairment [10]. In a recent publication [11] the authors propose a formal tool to explore the main design aspects of ADMIs based on Dimension Space Analysis. Such a form of analysis, a well-established methodology in the literature on interfaces for musical expression, aims at an effective visual representation of the design space.

A relevant ADMI category is based on tangible user interfaces (TUIs). This type of interface offers an alternative to graphical user interfaces, commonly used in information systems, by incorporating physical objects into interaction mechanisms. Such an approach gives digital information a tangible form, providing both representation and control capabilities. With TUIs, users can manipulate digital data using their hands and engage with them through their senses. The scientific literature reports several examples of music TUIs. For example, Paradiso *et al.* focused on the adoption of magnetic tags [12], Newton-Dunn *et al.* described how to control a dynamic and polyrhythmic sequencer using physical artifacts [13], and Schiettecatte and Vanderdonckt presented a distributed cube interface for sound design [14]. A successful case of commercially available music TUI is the *Reactable* [15], a backlit translucent tabletop surface equipped with a grid of LEDs and an underlying camera-based tracking system that recognizes specially designed physical objects and combines them with real-time digital audio processing and visual feedback. Finally, it is worth mentioning several approaches based on building blocks, e.g. LEGO bricks, to control musical parameters [16–18]. In general, music TUIs have been studied as alternative ways to control music parameters, mainly to foster creativity via embodiment or support music education. Nevertheless, they have been analyzed also in the context of ADMIs, for example for musical co-creation in families with impaired children [19] and for cognitive and motor rehabilitation of vulnerable users [20].

Another ADMI category utilizes gaze-based control [21]. This approach is, in a certain sense, complementary to the previous one as it is mainly directed towards users with motor difficulties. The possibility of tracking eye movements and detecting other facial gestures opens a number of ways to control the sound and music parameters. Many experiments have been conducted in this field [22–24]. Unfortunately, the poor capabilities of standard webcams push toward specialized hardware, i.e. eye trackers, which are expensive and not commonly found in computer systems. Hopefully, this limitation will be overcome by future technical advancements.

Nowadays, more attention is also paid to usability aspects within music-oriented software not specifically designed for impaired users. An example of this emerges from the

[2] https://www.musicasenzaconfini.com/
[3] https://audiomodeling.com/
[4] https://www.lim.di.unimi.it/

Figure 1. An ensemble music session involving a musician with impairment using a crafted version of *Inclusive MIDI Controller*, a guitarist, and a singer.

work by Pedrini *et al.* [25], which addresses the problem of evaluating the accessibility of digital audio workstations for blind or visually impaired people. The paper also contains the results achieved with three popular DAWs, namely *Cockos REAPER*, *Avid Pro Tools*, and *Steinberg Cubase*.

3. INCLUSIVE MIDI CONTROLLER

Inclusive MIDI Controller aims to be a versatile software application designed to facilitate MIDI control on multiple platforms. Its design was born from the idea of one of the present authors, founder of *Musica Senza Confini*. During his work on accessible ensemble music, he needed to find musical performance strategies for a user capable of moving only the fingers, toes, and eyes. Such a user-tailored solution included the use of the digital audio workstation (DAW) Logic PRO, the visual programming language for music and multimedia MAX Msp [5], and a manual setting of each track of the songs to be played to allow non-ordinary interactions. As shown in Figure 1, the user was able to play the controller in an ensemble, together with a guitarist and a singer. Specifically, the user could exploit an eye tracker to select the colored buttons on the screen and micro-muscular sensors to trigger controllers that were mapped on notes, chords, short MIDI sequences, or other sounds.

The next goal was to transform this crafted solution into a customizable software able to adapt to the user's characteristics and offering several pieces to play. This was the origin of the *Inclusive MIDI Controller* project. A schematic example of the proposed system is shown in Figure 2: *Inclusive MIDI Controller* runs on the computer in the middle, is controlled by a number of different input devices, and sends MIDI messages to a DAW running on another system. A multitrack project is loaded into the DAW, and each track in the project can be played by one of the connected devices, thus allowing collaborative music performance.

[5] `https://cycling74.com/products/max`

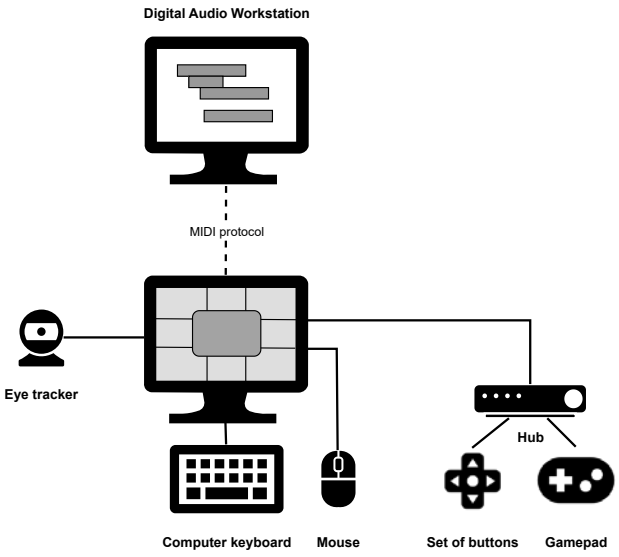


Figure 2. Scheme of the proposed system.

Programmed in C++ and leveraging the *JUCE* framework, [6] *Inclusive MIDI Controller* was designed to offer a seamless experience for both macOS and Windows users. As shown in Figure 4, *Inclusive MIDI Controller* basically provides a standalone interface equipped with a series of colored pads (whose number can be set from a minimum of 2 buttons per screen to a maximum of 12 buttons per screen), each capable of activation through standard or special input devices, thus facilitating accessibility to music and sound control.

The Settings menu allows the user to configure each colored button, and in particular:

1. the event to associate (MIDI Note, MIDI Control Change, Chord, MIDI Sequence, or page to select);
2. the behavior of the event (*Trigger*, i.e. sound activated on mouse hover; *Hold*, i.e. sound activated on mouse hover and deactivated when the mouse leaves the area; *Latch*, i.e. sound activated on mouse hover and deactivated by next mouse hover);
3. the MIDI channel;
4. the MIDI note;
5. the velocity value;
6. the release-velocity value;
7. a delay in milliseconds to apply to MIDI Note-On and Note-Off messages;
8. the color of the button;
9. the first and second keyboard shortcut;
10. an additional keyboard shortcut to repeat the latest note, which provides an aiding tool for some user categories (e.g., eye-tracker users).

[6] *JUCE* is an open source and C++ codebase framework for audio application and plug-in development. Web site: `https://juce.com`

Figure 3. The interface to configure settings.

Figure 4. Layout of the *Inclusive MIDI Controller* interface.

The Aux Pads menu presents further buttons, identical in appearance and configuration to the colored ones, but visible only through this specific menu. They can be activated via keyboard shortcuts only. The screen to set the parameters listed above is shown in Figure 3. The interface allows the user to save each setting locally, load it later, and reset each pad. The latest configuration is saved for future sessions.

Upon activation, these pads trigger the generation of MIDI messages received by other MIDI devices in the setup (e.g., a synthesizer or a DAW). The way such controls are activated depends on the hardware equipment (Human Interface Devices, HIDs) or software aiding tools in use on the user's system. The idea later explored in the observational study (see Section 4) is to test the accessibility of some functions of *Inclusive MIDI Controller* when this tool is matched with the input devices users are accustomed to.

One of the key features of this application is the integration of MIDI. Being a standard communication protocol, the MIDI messages generated by the system can be seamlessly routed to a Digital Audio Workstation (DAW), a plugin host, a music application, or even a hardware device, thus enhancing workflow flexibility and compatibility. To establish MIDI connections, *Inclusive MIDI Controller* adopts platform-specific methods. On macOS, the application exploits the virtual MIDI ports provided by CoreMIDI, ensuring communication with MIDI-enabled software and hardware. Conversely, on Windows systems, users may need to install additional drivers such as LoopBe [7] or LoopMIDI [8] to facilitate MIDI connectivity.

The cross-platform nature of the application, coupled with its interface and MIDI capabilities, makes *Inclusive MIDI Controller* a tool for musicians, producers, and audio enthusiasts seeking enhanced control and flexibility in their MIDI workflows. This software permits users to express their creativity triggering samples, controlling virtual instruments, or manipulating effects.

[7] https://nerds.de/en/loopbe1.html
[8] https://www.tobias-erichsen.de/software/loopmidi.html

4. OBSERVATIONAL STUDY

In this section, we will describe the methodology adopted to carry out the observational study. An observational study is an empirical investigation that attempts to estimate the effects caused by a process when it is not possible to perform an experiment [26]. This phase of the project is currently in progress and the end is scheduled for December 31, 2024.

Regarding the categories of individuals involved in the research, the following roles have been identified:

- Participant – An individual aged 18 or older;
- Supervisor – A designated individual collaborating on the research project.

For experimentation, each participant is associated with a numerical identifier whose purpose is to link the various data collected to the same participant. Associations between participants' first names and their respective numerical identifiers are stored only in paper format and are securely kept by the data controller. The personal data gathered from each participant are listed in Table 1. They are not expected to be disclosed or transferred to third parties. If this need emerges in the future, such data will be made strictly anonymous and/or presented in an aggregated form.

4.1 Research Question

The research question (RQ) and the hypothesis (H) underlying this observational study are the following:

- RQ — Can *Inclusive MIDI Controller* be understood and used by people with different abilities?
- H — There is no specific user target for *Inclusive MIDI Controller*. Each user can configure the software to employ the physical devices that she already knows or finds most congenial.

Figure 5. Partecipant 1's expanded keyboard with keyguard and mouse with joystick.

The basis of this question is the perspective that led to the design and implementation of *Inclusive MIDI Controller* as an accessible tool to include the largest part of individuals, overcoming their possible physical, cognitive, or social impediments.

4.2 Research Protocol

The research protocol consists of the four phases detailed below.

Phase 1: Initial questionnaire and instrument setup — The supervisor collects the participant's personal data through a specific questionnaire. During the administration of the questionnaire, the supervisor takes note of any additional comments from the participant. Subsequently, the supervisor configures the devices required for the study, including standard equipment (a computer and an audio interface) and, potentially, user-specific input devices. The software installed on the computer is the DAW *Ableton Live*, *Inclusive MIDI Controller*, and *Open Broadcaster Software*, a free and open source software for video recording and live streaming. In general, the computer equipment and software are provided by the supervisor, but users are requested to bring their own hardware tools in case of special needs.

Phase 2: Training — During the training phase, the participant is guided and assisted in choosing the hardware device to use for the observational study. If the participant is already used to a specific tool and such a tool is available among the options, it will be employed; otherwise, the participant is asked to bring their own device or assistance will be provided in selecting an appropriate tool for the purpose. The same applies if the participant has never used any specific hardware tool before. Next, the software is presented, and its purpose and usage are explained. However, further details about the controls offered by the software are not provided to participants, as the test focuses on the user experience and aims to assess the design choices.

Phase 3: Test — During the test phase, the participant is asked to perform certain actions on the software without prior knowledge of its structure. The exercises planned during the study are shown in Table 2. During the testing phase, monitoring tools will be used to enable data collection [27]. Activity logging will be implemented within the software used for the test to track user activities on it. Additionally, an audio (microphone) video and screen recording tool OBS (Open Broadcaster Software) will be utilized. Notes will be taken for each participant.

Phase 4: Questionnaire — At the end of the test phase, a questionnaire is administered to participants to investigate the perceived experience and gather their feedback. The questionnaire includes some closed responses on a 5-point Likert scale, where 1 implies "very little" and 5 means "very much," as well as some open responses. The questions are shown in Table 3.

4.3 The Test

4.3.1 Participant Descriptions

Three users participated in the test activity, each with unique characteristics related to their age, sex, impairments, musical knowledge, musical experience, and tools used for computer interaction and playing digital musical instruments. The profiles of the participants are summarized in Table 4.

Participant 1 (P1) is a woman aged between 18 and 27 years old, who faces the challenge of quadriplegia. She declares an above-average level of musical experience and an average level of musical knowledge. Her computer interaction tools include an expanded keyboard with a keyguard and a mouse with a joystick (see Figure 5), facilitating her engagement with digital interfaces. For playing musical instruments, she uses a pedal lift for drums.

Participant 2 (P2), a man aged between 48 and 57 years old, experiences a light form of daltonism, a color vision deficiency. He possesses a good level of musical knowledge and an average level of musical experience. For computer interaction, he relies on conventional tools such as a mouse, keyboard, and trackpad, indicating proficiency in navigating digital interfaces. Although he does not use specific tools to play digital musical instruments, his participation provides insight into the experiences of people with light visual impairments in music-related activities.

Finally, Participant 3 (P3), a man between 38 and 47 years of age, faces the challenge of total blindness. He declares an average level of musical knowledge and musical experience. His computer interaction tools include a screen reader and a braille display, enabling him to access digital content and interfaces through auditory and tactile feedback. He does not utilize specific tools for playing musical instruments.

Participants in our observational study have been chosen to represent a diverse range of ages, genders, impairments, musical backgrounds, and technological adaptations, thus providing valuable perspectives for understanding accessibility and inclusivity in digital music activities.

4.3.2 Test Results

This section provides a description of the results obtained by participants P1, P2, and P3 in achieving the tasks listed in Table 2.

Table 1. User Profiling.

Personal Data	Response Type
Age	Range: "18-27", "28-37", "38-47", "48-57", "58-67", "68-77", "78-87", "88-97"
Gender identity	"Man", "Woman", "Transgender", "Non-binary/non-conforming", "Prefer not to respond"
Health conditions or impairments	Open-ended response
Musical knowledge	"Little or none", "Sufficient", "Average", "Good", "Excellent"
Musical experience	"Little or none", "Sufficient", "Average", "Good", "Excellent"
Tools used for computer interaction	Open-ended response
Tools used for playing digital musical instruments	Open-ended response

Table 2. Tasks for the Test Phase.

Level	Task	Description
Base	1	Press Button 1
Base	2	Maximize the window to full screen
Intermediate	3	Access settings for Button 3
Intermediate	4	Set the number of pads to 6 for page 1
Advanced	5	For AuxPad 1, set Event to "note", Type to "trigger", Key 1 to "cursor right", and Key 2 to "cursor left"
Advanced	6	Configure Pad 2 to navigate to Page 2

Table 3. Post Test Questionnaire.

Question	Response Type
How easy was the use of the software?	Closed response
How pleasant was the use of the software?	Closed response
How frustrating was the understanding of the software?	Closed response
How frustrating was the use of the software?	Closed response
What feelings or thoughts predominated your mind during the exercise?	Open-ended response

Table 4. Participant Profiles.

User	Age	Gender	Impairments	Musical Knowledge	Musical Experience	Tools for Computer Interaction / Musical Instruments
P1	18-27	Woman	Quadriplegia	Average	Good	Expanded keyboard with keyguard, mouse with joystick / Pedal lift for drums
P2	48-57	Man	Daltonism	Good	Average	Mouse, keyboard, trackpad / N/A
P3	38-47	Man	Blindness	Average	Average	Screen reader, braille display / N/A

Regarding *Task 1*, P1 successfully moved the joystick over Button 1 of *Inclusive MIDI Controller*, producing the expected sound. However, when attempting to play the button by typing Key 1 on the keyboard, this action did not yield the desired result. P2 executed the action correctly and swiftly. Finally, due to his specific impairment, *Task 1* was not testable for P3.

Focusing on *Task 2*, all participants performed the action accurately and promptly. However, P2 encountered a challenge when attempting to click the button used for enlarging a window to full screen, as it was disabled. Instead, he resorted to dragging the edges of the window until they reached full-screen mode. Finally, he noticed the dedicated "full-screen" button.

For *Task 3*, P1 thoughtfully moved the cursor to Button 3 of *Inclusive MIDI Controller*and activated it, she attempted to access the settings but seemed unable to find the desired options. Eventually, she located the sidebar with the list of buttons and accessed the settings for Button 3. Concerning P2, first, he unsuccessfully attempted to access the settings directly from the button itself, then he clicked the Option button in order to explore the possibilities offered, once again not finding what he was looking for, and he finally hit the Settings button, where he managed to perform the requested action. On the contrary, P3 performed the task accurately and quickly.

Regarding *Task 4*, all participants successfully opened the Page drop-down menu, even though the selected page was already correct. P3 encountered a challenge as the screen reader did not read the drop-down menu labels. Nevertheless, he completed the task swiftly, drawing on previous software experience.

P1 and P2 executed the action required by *Task 5* correctly and swiftly. P3 entered the correct settings, but he faced difficulty completing the task due to the screen reader not reading the labels of the drop-down menus. This problem also emerged during Task 4. The correct setting for *Task 5* is shown in Figure 3.

Finally, regarding *Task 6*, P1 was able to act correctly and swiftly. She only had to take a brief pause to figure out where to locate the command to navigate to a particular page. P2 initially attempted to set AuxPad 2 but corrected himself independently and was able to complete the task. Once again, P3 faced a similar challenge with the screen reader not reading drop-down menu labels, hindering task completion.

4.3.3 Post-Test Questionnaire

At the end of the test phase, a short questionnaire was administered to the participants to assess some aspects of the user experience (see Table 3). For closed responses, a 5-point Likert scale was adopted.

P1 reported a score of 4 for the ease of use and a score of 5 for the fun level. She expressed a level of frustration of 2 both in understanding and in using the software. She found the interface to be cute and fun, expressing a desire to use it again.

P2 reported a score of 5 for the ease of use and 2 for

the fun level. He expressed a frustration level of 1 both in understanding and in using the software. In response to the last question, P2 remarked that he expected a behavior similar to his operating system, finding conversely some buttons unintuitive in their purpose.

Finally, P3 reported a score of 2 for the ease of use and a score of 3 for the fun level. He expressed a frustration level of 2 in understanding the software and a frustration level of 3 in using the software. For certain impairments, P3 experienced difficulty in discerning the current location within the interface and understanding the outcome of specific actions. He noted a lack of feedback following actions, leading to uncertainty about changes in state or context. These aspects resulted in an above-average level of frustration in using the software.

5. CONCLUSIONS

In this work, we presented a project addressing a potentially wide audience, without exclusion in the face of possible disabilities or impairments, and dealing with accessibility and inclusion in musical expression. The first step in evaluating the platform was to test the ability of heterogeneous users to complete selected tasks of increasing difficulty levels. In order to assess our approach, an extensive test campaign should be conducted. Unfortunately, finding participants with the required profile is not a trivial task. Indeed, to have a comprehensive overview, users should have various types of cognitive and physical disabilities, use heterogeneous input devices and tools, possess musical interests, and have diversified levels of musical skills. For this reason, we chose to carry out an observational study, as often proposed in the literature for collaborative interfaces [28]. Such a study gave us the possibility to observe the usability of *Inclusive MIDI Controller* perceived by different types of real users and gather some quantitative data to drive future improvements.

In particular, we implemented a prototype version of *Inclusive MIDI Controller* and conducted our observational study involving three users. The main advantage of such an approach is the ease of observing the effect of a specific variable as it occurs naturally, without making any attempt to intervene. Despite the limited number of responses, which cannot be considered representative of the entire target population, this study allowed us to validate the design and highlight some limitations.

The results collected so far reflect heterogeneous levels of success and challenges encountered by participants with different impairments. In general, most of the time, users were able to complete the assigned tasks in a short time. Anyway, some flaws emerged in the accessibility of interface elements for blind people, an issue that needs to be addressed before releasing the platform.

In the evaluation of the results, it is important to note that it was a blind test and that the participants had no previous training on the interface. A positive aspect emerging from the post-test questionnaire is the low level of frustration in understanding and using the software. In this sense, *Inclusive MIDI Controller* seems to be a promising tool in the context of accessible collaborative music performances.

The next steps will be the implementation of corrective actions to respond to user comments and observations, the organization of a campaign with a larger number of participants, and the field testing in a real-world scenario by connecting the controller to a music-making system.

6. REFERENCES

[1] M. Harniss, *Accessibility*. Dordrecht: Springer Netherlands, 2014, pp. 9–10. [Online]. Available: https://doi.org/10.1007/978-94-007-0753-5_8

[2] A. Kavcic, "Software accessibility: Recommendations and guidelines," in *International Conference on Computer as a Tool (EUROCON)*, vol. 2. IEEE, 2005, pp. 1024–1027.

[3] Mozilla, "Software accessibility: Where are we today?" https://developer.mozilla.org/en-US/docs/Web/Accessibility, 2019, accessed: 2024-03-11.

[4] L. Nummenmaa, V. Putkinen, and M. Sams, "Social pleasures of music," *Current Opinion in Behavioral Sciences*, vol. 39, pp. 196–202, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352154621000747

[5] E. Steinfeld and G. S. Danford, "Theory as a basis for research on enabling environments," in *Enabling Environments*. Springer, 1999, pp. 11–33.

[6] S. Iwarsson and A. Ståhl, "Accessibility, usability and universal design—positioning and definition of concepts describing person-environment relationships," *Disability and rehabilitation*, vol. 25, no. 2, pp. 57–66, 2003.

[7] M. Manzoor and V. Vimarlund, "Digital technologies for social inclusion of individuals with disabilities," *Health and technology*, vol. 8, no. 5, pp. 377–390, 2018.

[8] N. Davanzo and F. Avanzini, "Hands-free accessible digital musical instruments: Conceptual framework, challenges, and perspectives," *IEEE Access*, vol. 8, pp. 163 975–163 995, 2020.

[9] E. Frid, "Accessible Digital Musical Instruments—A Review of Musical Interfaces in Inclusive Music Practice," *Multimodal Technologies and Interaction*, vol. 3, no. 3, p. 57, Jul. 2019.

[10] R. B. Knapp and H. S. Lusted, "A bioelectric controller for computer music applications," *Computer Music Journal*, vol. 14, p. 42, 1990. [Online]. Available: https://api.semanticscholar.org/CorpusID:62767345

[11] N. Davanzo, F. Avanzini *et al.*, "A dimension space for the evaluation of accessible digital musical instruments," in *Proceedings of the International Conference on New Interfaces for Musical Expression*. Birmingham City University, 2020, pp. 214–220.

[12] J. A. Paradiso, K.-y. Hsiao, and A. Benbasat, "Tangible music interfaces using passive magnetic tags," in *Proceedings of the 2001 Conference on New Interfaces for Musical Expression*, ser. NIME '01. SGP: National University of Singapore, 2001, p. 1–4.

[13] H. Newton-Dunn, H. Nakano, and J. Gibson, "Block jam: A tangible interface for interactive music," *Journal of New Music Research*, vol. 32, no. 4, pp. 383–393, 2003.

[14] B. Schiettecatte and J. Vanderdonckt, "Audiocubes: A distributed cube tangible interface based on interaction range for sound design," in *Proc. 2nd Int. Conf. on Tangible and Embedded Interaction*, ser. TEI '08. New York, NY, USA: Association for Computing Machinery, 2008, p. 3–10. [Online]. Available: https://doi.org/10.1145/1347390.1347394

[15] S. Jordà, G. Geiger, M. Alonso, and M. Kaltenbrunner, "The reacTable: exploring the synergy between live music performance and tabletop tangible interfaces," in *Proc. 1st Int. Conf. on Tangible and Embedded Interaction*, 2007, pp. 139–146.

[16] A. Baratè, L. A. Ludovico, and D. Malchiodi, "Fostering computational thinking in primary school through a LEGO®-based music notation," *Procedia Computer Science. Knowledge-Based and Intelligent Information & Engineering Systems: Proceedings of the 21st International Conference, KES 2017, 6-8 September 2017, Marseille, France*, vol. 112, pp. 1334–1344, 2017.

[17] K. Gohlke, M. Hlatky, and B. de Jong, "Physical construction toys for rapid sketching of tangible user interfaces," in *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction*, ser. TEI '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 643–648. [Online]. Available: https://doi.org/10.1145/2677199.2687900

[18] B. Miotti, L. Bassani, E. Cauteruccio, and M. Morandi, "MusicBlocks: An innovative tool for learning the foundations of music," in *Proceedings of CSME 2022*, 2022, pp. 475–484.

[19] K. Stensaeth, "Musical co-creation? Exploring health-promoting potentials on the use of musical and interactive tangibles for families with children with disabilities," *International Journal of Qualitative Studies on Health and Well-being*, vol. 8, no. 1, p. 20704, 2013.

[20] A. Baratè, H. Korsten, L. A. Ludovico, and E. Oriolo, "Music tangible user interfaces and vulnerable users: State of the art and experimentation," in *Computer-Human Interaction Research and Applications. 5th International Conference, CHIRA 2021, and 6th International Conference, CHIRA 2022, Revised Selected Papers*, ser. Communications in Computer and Information Science, L. Constantine, A. Holzinger, H. P. Silva, and J. Vanderdonckt, Eds. Springer Science and Business Media Deutschland GmbH, 2023, vol. 1882, pp. 1–25.

[21] N. Davanzo, F. Avanzini *et al.*, "Design concepts for gaze-based digital musical instruments," in *Proceedings of the 19th Sound and Music Computing Conference*. SMC network, 2022, pp. 472–478.

[22] Z. Vamvakousis and R. Ramirez, "The eyeharp: A gaze-controlled digital musical instrument," *Frontiers in psychology*, vol. 7, p. 198177, 2016.

[23] D. Kandpal, P. R. Kantan, and S. Serafin, "A gaze-driven digital interface for musical expression based on real-time physical modelling synthesis," in *19th Sound and Music Computing Conference, SMC 2022*. Sound and Music Computing Network, 2022, pp. 461–468.

[24] N. Davanzo, L. Valente, L. A. Ludovico, and F. Avanzini, "Kiroll: A gaze-based instrument for quadriplegic musicians based on the context-switching paradigm," in *Proceedings of the 18th International Audio Mostly Conference*, 2023, pp. 59–62.

[25] G. Pedrini, L. A. Ludovico, and G. Presti, "Evaluating the accessibility of digital audio workstations for blind or visually impaired people," in *Proceedings of the 4th International Conference on Computer-Human Interaction Research and Applications (CHIRA 2020)*, L. Constantine, M. Helfert, A. Holzinger, and H. P. Silva, Eds. SCITEPRESS - Science and Technology Publications, Lda., 2020, pp. 225–232.

[26] P. Rosenbaum, *Observational Study: Definition and Examples*. Wiley, 09 2014.

[27] J. Lazar, J. H. Feng, and H. Hochheiser, Eds., *Research Methods in Human Computer Interaction*, 2nd ed. Boston: Morgan Kaufmann, 2017.

[28] T. Alrashed, A. Almalki, S. Aldawood, T. Alhindi, I. Winder, A. Noyman, A. Alfaris, and A. Alwabil, "An observational study of usability in collaborative tangible interfaces for complex planning systems," *Procedia Manufacturing*, vol. 3, pp. 1974–1980, 2015, 6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015.

SINGING FOR ANXIETY: EXPLORING THE POTENTIALS OF SINGING IN IMMERSIVE VIRTUAL REALITY TO HELP CHILDREN WITH SOCIAL ANXIETY

Ali ADJORLU (adj@create.aau.dk) (0000-0001-5414-8631)[1] and
Stefania SERAFIN (sts@create.aau.dk) (0000-0001-6971-1132)[1]

[1]*Architecture Design and Media Technology*, **Aalborg University**, A. C. Meyers Vænge 15, 2450 Copenhagen, Denmark

ABSTRACT

This paper presents an immersive Virtual Reality application designed for exposure therapy on children suffering from social anxiety. Two different interfaces for communication between the therapist and the client are designed and evaluated: One allowing a therapist to be immersed in the virtual environment with the client while wearing an HMD (3D user interface) and one where the therapist uses a traditional screen with only the client wearing an HMD (2D user interface). The therapist can use the interface to change the virtual scene in order to gradually expose the user to social singing situations in order to reduce anxiety.

A heuristic evaluation and a focus group interview were conducted on four clinicians, all experienced on working with children with social anxiety on a daily basis. According to the therapist participating in this explorative study, being immersed in VR and wearing an HMD together with the client increases the chances of creating trust and collaboration between the child and the therapist, while the traditional UI on a desktop screen is easier to use compared to the 3D UI.

1. INTRODUCTION

Social Anxiety Disorder (SAD) refers to the fear of being negatively judged by others in social situations, resulting in significant negative consequences on the life quality of individuals suffering from it [1].

Anxiety disorders are the most common psychiatric disorders among children and adolescents. A meta-analysis estimated the prevalence of anxiety disorders to be 12.3 % for children (6-12) and 11% for adolescents (13-18) [2].

Children with SAD on average, get lower grades and a higher degree of school drop-out than those without the disorder [2]. Upon adulthood, underemployment and underpaid jobs are more common among individuals suffering from SAD, making them more financially dependent [3]. Additionally, loneliness is more common among individuals with SAD due to difficulty getting into close relationships with others. These deficiencies underline the importance of interventions to help them cope with or reduce their anxiety.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

One of the most effective methods to treat anxiety is exposure therapy [4]. By gradually presenting the patient with stimuli that provoke their anxiety, the clinician seeks to remove the negative emotions through guidance and repetition. Exposure therapy for social anxiety typically involves exposing the patient to real-life social situations, such as giving a speech in front of an audience or attending a party. The psychologist will often be present with the patient, correcting his maladaptive behavior and beliefs and providing appropriate coping strategies.

To our knowledge, exposure therapy was first conducted for social phobia in the mid-1980s [5]. This relatively late introduction is because exposure to realistic social situations is challenging to conduct and organize. Organizing exposure therapy for SAD requires people and physical space to create a socially anxiety-inducing scenario, requiring many resources.

By substituting real-world sensory information with digitally generated ones, virtual reality (VR) offers controllable, interactive, relevant virtual environments for exposure therapy [6]. Additionally, VR enables clinicians to perform exposure therapy in environments and contexts that are otherwise not easily accessible [7] [8].

Back in 1994, North et al. published one of the first known studies on using virtual reality to perform exposure therapy for individuals who suffer from acrophobia [9]. Since this pioneering study, a vast body of research has looked into using VR as a tool to treat a variety of anxieties, such as aerophobia [10], acrophobia [11], claustrophobia [12], and social anxieties [13]. A recent systematic review indicates that, in general, VR-based exposure therapy interventions for social anxiety have similar efficiency as in vivo exposure therapy [14]. As an example, a study conducted by Bouchard et al. on 59 participants showed that VR exposure therapy resulted in similar improvements in the Liebowitz social anxiety score as in-vivo exposure therapy [15].

Singing, and especially singing in a choir, can have psychological benefits for individuals with chronic mental health issues, with research indicating a reduction in anxiety and an increase in positive emotions and relaxation, which translates to maintained and extended mental well-being [16, 17]. Music, as a universal language that transcends the barriers of verbal communication, provides a medium through which children can express emotions, build confidence, and foster social connections in a non-threatening environment. Research has demonstrated that engaging in musical activities, such as singing or playing

an instrument, can significantly decrease levels of social anxiety in children by enhancing their self-esteem and social skills. Moreover, the structured yet flexible nature of music therapy allows for tailored interventions that meet the individual needs of children with social anxiety, encouraging gradual exposure to social situations in a controlled and supportive setting. However, research on the benefits of singing in virtual reality is in its infancy, with an acknowledged need for robust interdisciplinary studies [18].

Incorporating VR technology into music therapy presents a novel approach to treating social anxiety in children. VR-based musical interventions can simulate real-life social situations in a virtual environment, providing a unique opportunity for children to practice and enhance their social skills without the immediate pressures of actual social interactions.

Unfortunately, despite the benefits of VR as a tool for exposure therapy on individuals with SAD, no studies have been reported in recent systematic reviews on the topic with children as the primary target group for the intervention [14, 18]. In this paper, we present a VR intervention designed to perform exposure therapy for children with social anxiety by placing them in a virtual environment where they get to sing together with or in front of the Danish National Children's Choir.

This paper presents a VR intervention designed to be a tool for therapists working with children suffering from social anxiety. However, this brief research report will focus on the needs and requirements of psychologists and therapists working with children and adolescents diagnosed suffering from SAD.

Brinkman et al. conducted a study focusing on the needs and requirements of the therapists rather than the client during a VR exposure therapy session [19]. This study resulted in a set of design guidelines when designing user interfaces (UI) for therapists to control a VR exposure therapy session. These guidelines focus on allowing the therapist to effortlessly control the VR scenarios experienced by the client and help them avoid activating unwanted events in the simulation. The design guidelines were for traditional screen-based UI. All of the studies described in the systematic review on VR for SAD conducted by Horigome et al. focus on the outcome of the users suffering from SAD [14]. However, several of these studies describe an interface designed for the therapist to control variables in the VR SAD exposure therapy interventions, such as the virtual avatars' number of- and reactions. These interfaces utilize traditional interfaces such as a mouse and keyboard for the therapist to control the VR environment while only the clients are immersed in VR wearing a head-mounted display (HMD). Some scholars suggest that having the client wear an HMD can negatively affect the alliance between the client and the therapist due to the lack of eye contact [20]. This poses the question: what if the therapist is co-present in the virtual environment with the client while wearing an HMD and controlling an avatar, resulting in eye contact and gestural interaction in the virtual environment.

This paper presents a VR SAD exposure therapy intervention with a 3D UI interface designed for the therapist to use while controlling a 3D avatar and immersed via an HMD in the same virtual environment as the client. This intervention is then compared to a traditional UI controlled with a mouse and keyboard on a 2D screen to shed some initial light on the pros and cons of having therapists immersed in VR with their clients during exposure therapy.

2. METHODS

2.1 The VR intervention

The VR exposure therapy application consisted of eight 360° recordings of the Danish National Children Choir singing in one of their rehearsal halls at the Danish National TV studios (see figure 1).

Figure 1. *Top: how the child sees the 3D avatar of the psychologist. Bottom: Singing session recorded from the second row.*

A total of 25 children and one conductor participated in the recording sessions. In each video, the camera, the children, and the conductor's locations were changed to capture scenes with varying levels of anxiety-inducing stimuli (see fig: 2).

- A scene where the camera is placed in the second row behind the rest of the choir members to create a socially safe VR experience.)

- A scene where the camera is placed in the first row creates a more anxiety-inducing experience than the previews scene.
- A scene where the camera is in front, next to the conductor, facing all of the children in the choir, creating the most anxiety-inducing experience.
- Three scenes were recordeded where the nearest choir member's distance from the camera are 40 cm, 60 cm, and 100 cm, to give the psychologist the option to create a different type of anxiety-inducing experience.

To explore whether the 360 recordings had the potential to be used as the main platform for the VR SAD exposure therapy intervention, we conducted a formative qualitative evaluation with two psychologists and one psychology student working in a clinic for children with anxiety. The psychology student is suffering from social anxiety. All three

Figure 2. *From top left to right: 1- The camera is placed in the first row 2- The camera is placed in the second row behind the rest of the choir members 3- The camera is placed next to the conductor facing all of the children in the choir.*

participants experienced the VR singing videos, followed by an unstructured group interview to get their views and suggestions for the intervention's next iteration.

During the post-exposure interview, the psychology student with social anxiety stated that she felt a varying level of social anxiety having to sing in the back row compared to having to sing in front of all other choir members. She felt judged by the other choir members when placed facing them all and asked to sing. These statements from the psychology student do, to some extent, confirm that the videos might potentially be useful as a tool for exposure therapy on children diagnosed with social anxiety.

Both psychologists stated the importance of being able to change the amount of anxiety induced in the VR intervention gradually depending on the state of the client's mind. They believed that the videos potentially could provide a varying degree of anxiety. They stated that since each child is different, they needed individualized treatment based on their specific needs, anxieties, and state of mind during the exposure therapy session. Changing scenes, pausing, fast-forwarding, or rewinding the 360 videos were among the psychologists' requests to control the VR exposure therapy intervention.

Additionally, the psychologists mentioned the importance of trust and alliance between the child and the therapist during the exposure therapy session. The psychologists stated that the child should be able to point out anxiety-inducing elements in the virtual environment.

Based on this feedback, the next iteration of the application was developed. Three main functional requirements were identified:

- The ability to change the scenes in the virtual environment.
- The ability to pause, play, rewind, or fast-forward the 360 video.
- The psychologist should be able to communicate with the child using verbal and gestural interaction.

Additionally, in order to explore whether the alliance and communication between the psychologist and the client can benefit from having the psychologist immersed in the same virtual environment as the child, two versions of the interface were created (see figure 3):

1. *HMD Interface*: The psychologist is present in the virtual environment with the child, controlling a 3D avatar via HMD and VR controllers.
2. *Desktop interface*: The psychologist is present in the virtual environment with the child controlling a 3D avatar using the keyboard and mouse on a desktop computer (see figure 3).

The child will log in to the virtual environment using an Oculus HMD and touch controllers on both versions of the VR intervention. The application and the networking protocols are implemented in the game engine 3D Unity.

The psychologist and the child can control their 3D avatar and 3D hand models, both of which are communicated over the network. This allows the psychologist and the child to see each other's avatar body gestures in real-time. Only the psychologist can control the 360 video being played in the scene using either a 3D UI via the HMD interface or a traditional GUI in the desktop interface. These interfaces allow the psychologist to change the 360 video being played, pause, unpause, fast forward, and rewind. Additionally, the UI allows the psychologist to disable his avatar, giving the client the illusion of being alone in the virtual environment if needed.

2.2 Evaluation

Two psychologists and two psychology students, all working at a clinic for children suffering from social anxiety in Copenhagen, participated in an explorative evaluation of the application.

Participant A is an older male psychologist with many years of experience working with children and adolescents with mental disorders. He owns two clinics in Denmark, one in Copenhagen and one in Roskilde, both specialized in working with children with anxieties.

Participant B is a male psychologist working in the clinic owned by Participant A.

Participant C (Identifies as female) psychologist student working at a clinic owned by Participant A. She is experienced in working with children with social anxiety.

Figure 3. *Left: Singing session recorded from the second row. Right: how the child sees the 3D avatar of the psychologist*

Participant D is also a psychologist student and works as an assistant at the clinic owned by Participant A. Additionally, participant D suffers from a minor degree of social anxiety herself.

The study was conducted in the clinic owned by Participant A.

A within-group design with a predetermined shifted order for which UI version the participant should experience was conducted. Each participant tried both the desktop and the VR version of the application. A thorough introduction and walkthrough on how to use the interventions were given to each participant before they tried it themselves. Once they got to try the application themselves, one of the authors would log in to the app via the HMD on the second laptop and play the role of a child with social anxiety. The participants were given five minutes to try out the application with the author logged in as the child.

After five minutes, the author would provide the participants with a set of assignments. The assignments include changing to a specific scene, fast forward, pausing, unpausing, and making the 3D avatar invisible or visible to the child. Following the task, the participants were asked a set of questions based on the revised set of heuristics proposed by Jakob Nielsen [21]. The chosen heuristics were visibility of system status, user control and freedom, error prevention and flexibility, and efficiency of use. Following the heuristic questions, a focus group interview was conducted with all four participants after trying both the desktop and the VR versions of the intervention.

2.3 Results

2.3.1 Heuristic Evaluation

The first heuristic was on flexibility and efficiency of use. In regards to the VR version of the intervention, all four participants reported that the system allowed them to control the exposure therapy session and interact with the child efficiently and flexibly. However, three out of the four participants (A, B, and D) reported that they had some issues fast-forwarding and rewinding the 360 videos when using the 3D UI.

For the desktop version of the application, three out of the four participants (A, B and C) believed that the system was efficient and flexible in supporting them in controlling the exposure therapy session while communicating with the child. Participant D stated that she had a hard time exploring the scene using the mouse and keyboard.

The second heuristic focused on error prevention. Regarding the 3D UI version, all four participants answered that they might make some errors during prolonged interaction with the system due to their inexperience with VR.

In regards to the error prevention of the desktop version of the intervention, three participants stated that they did not believe that they would make any mistakes while performing exposure therapy with the application.

The third heuristic was on the visibility of the system status. For the VR version of the application, three participants mentioned that they spent too much time trying to read the labels on the buttons, while one participant was interested in having an indication of how far they were into the movie despite there being a timeline on the 3D UI.

All four participants stated that they had a good overview of the system status via the user interface for the desktop version. One participant mentioned that this is due to him being more used to interacting with user interfaces using the keyboard and mouse.

The final heuristic was on user control and freedom. In regards to the 3D UI version, all participants mentioned that the system provided them the freedom to explore the scene effortlessly.

Regarding the desktop edition of the application, all participants stated that they were in full control of the exposure therapy session. Performing actions such as changing scenes was easy for all participants. Participant B added that even though he got control and freedom with this version, he missed out on experiencing the environment on the same level as the child (while wearing an HMD). Participant D stated that she struggled to explore the scene using the keyboard and mouse.

2.3.2 The focus group interview

Traditional coding [22] of the focus group interview resulted in four main categories: weaknesses and strengths of the desktop version of the application and weaknesses and strengths of the VR version of the application. Each of

Figure 4. *Results of the focus group interview (Orange: Desktop UI. Blue: 3D UI)*

these categories includes several subcategories, which can be seen in figure 4.

There were a total of eight positive and seven negative comments on the VR version of the application. The most dominant topic among the positive comments was the power of VR to create equality between the psychologist and the child. Participant A stated that it could make the child feel safer if he knew that the psychologist is present in the virtual universe. Participant B added that it is crucial for a psychologist to put himself in the child's shoes to create empathy. This is easier to achieve when experiencing the intervention wearing an HMD. Participant B stated that sometimes the child does not know why they are afraid of certain situations. By experiencing the scenario in the same way as the child using the HMD, it can become more manageable for the psychologists to discuss the anxiety-triggering events and offer an appropriate guidelines to the child on how to deal with them. Another positive topic discussed by the participants was trust. Participant D stated that if the child sees the psychologist wear the HMD and enter the virtual environment, it can increase the child's trust towards the whole intervention.

Participant C stated that not being able to see parameters such as the child's facial expressions and body language was one of the negative sides of both versions of the UI. Participant A states that these parameters are vital when doing in-vivo exposure therapy, giving VR exposure therapy a major handicap compared to it. Finally, during the focus group interview, participant B mentioned on three occasions that he felt some level of cybersickness while wearing the HMD. All four participants had negative comments regarding the user-friendliness of the 3D UI.

In regards to the desktop version of the application, a total of five positive comments were made during the focus group interview. Participant B stated that it was easier for him to keep an eye on what the child is doing while he tried to manipulate the exposure therapy session using the UI on the desktop version. He would have to move his left hand in front of his face in the VR version, covering the whole scene while he tried to press a button using the oculus touch controllers. Participants A and B both stated that it is much easier for them to see how the child feels since they could see the child's body posture, skin color change, and sweat in real life when they were not wearing an HMD. Three out of the four negative comments regarding the desktop version of the application concerned the possibility of producing performance anxiety in the child if being the only one to wear an HMD. Participant D, who herself has some level of social anxiety, stated that when she was asked to wear the HMD, she felt nervous that she was the only one wearing the HMD and that others might negatively judge her. Finally, participants C and D stated that it was harder for her to navigate the scene via the desktop version than the application's VR version. Moving her head to look around seemed more natural to her than having to use the mouse and keyboard to explore the scene.

3. DISCUSSION AND CONCLUSION

In this paper, we set out to explore the benefits and disadvantages of allowing a psychologist to be immersed in the same virtual environment as the client when performing exposure therapy sessions. In order to do so, we created a SAD VR exposure therapy application with two interfaces for a therapist: One used while immersed in VR and one used via a traditional desktop computer.

During traditional in-vivo exposure therapy sessions, the psychologist is present with the client in the fear-inducing environment, observing the client's behavior while providing appropriate guidance and feedback. Therefore, psychologists often analyze facial expression, body postures, and other parameters to interpret the emotions of their clients and then provide appropriate input based on these parameters. However, not all of these visual parameters are available when the client wears an HMD covering the face. What if the psychologist is also wearing an HMD and joins the client in the same virtual environment? In our implementation, when wearing the HMD, the psychologists and therapists could only see a 3D avatar that moved based on their clients' head and hand movements. In general, according to the therapists participating in the short explorative study described in this paper, being immersed in VR enables the psychologist to create a greater sense of trust and collaboration with their client compared to when controlling the intervention via a desktop. Additionally, the VR version allows the client and the therapist to have eye contact in the virtual world, which is not possible when only the client is wearing an HMD. However, when the psychologist is wearing an HMD, some real-life information will be hidden, such as changes of skin color and sweat due to anxiety. In future iterations, biometric data can be provided to the therapist in real-time using devices such as the Empathica wristband [23] to increase the therapists' ability to analyze the clients' reactions while both are immersed in the intervention. This information can be intuitively presented to the therapist in the virtual environment.

The participants were, in general, more positive towards

the UI on the desktop version of the application compared to the UI for the VR version of the application. Previous experience using UI on desktop computers using the keyboard and mouse is the main reason for this preference. It can be argued that after some time of exposure and experience with interacting with user interfaces while wearing an HMD can increase the psychologists' abilities in using a UI while wearing an HMD. However, there is still much work to be done when designing user interfaces to be used while wearing an HMD. As an example, due to the low resolution of the Oculus Rift HMD, it is not possible to read the labels on the buttons unless the UI is moved very close to cover most of one's field of view. Newer HMD's such as the Oculus Rift S or the HTC Vive Pro have a higher resolution that can fix this issue. However, future studies should investigate what methods to use to navigate and interact with graphical user interfaces while using VR controllers such as the oculus touch or the Vive Index. This study takes some essential initial steps towards understanding the needs of psychologists performing exposure therapy using VR instead of focusing on individuals suffering from social anxiety. Future research should further investigate the needs and requirements of psychologists and therapists when designing VR interventions for exposure therapy for children since they are one of the essential stakeholders.

Acknowledgments

We want to thank the Danish National Choir and the Danish National Radio for allowing us to record the sessions used in this paper. Additionally, we would like to thank Søren Benedikt Pedersen and his colleagues from "Cool Kids Copenhagen" for providing valuable feedback.

4. REFERENCES

[1] I. M. Aderka, S. G. Hofmann, A. Nickerson, H. Hermesh, E. Gilboa-Schechtman, and S. Marom, "Functional impairment in social anxiety disorder," *Journal of anxiety disorders*, vol. 26, no. 3, pp. 393–400, 2012.

[2] E. J. Costello, H. L. Egger, and A. Angold, "The developmental epidemiology of anxiety disorders: phenomenology, prevalence, and comorbidity," *Child and Adolescent Psychiatric Clinics*, vol. 14, no. 4, pp. 631–648, 2005.

[3] H.-U. Wittchen, M. Fuetsch, H. Sonntag, N. Müller, and M. Liebowitz, "Disability and quality of life in pure and comorbid social phobia–findings from a controlled study," *European Psychiatry*, vol. 14, no. 3, pp. 118–131, 1999.

[4] E. B. Foa and C. P. McLean, "The efficacy of exposure therapy for anxiety-related disorders and its underlying mechanisms: The case of ocd and ptsd," *Annual review of clinical psychology*, vol. 12, pp. 1–28, 2016.

[5] G. Butler, "Exposure as a treatment for social phobia: Some instructive difficulties," *Behaviour Research and Therapy*, vol. 23, no. 6, pp. 651–657, 1985.

[6] A. Adjorlu, "Virtual reality therapy," in *Encyclopedia of Computer Graphics and Games (ECGG)*. Springer, 2018.

[7] P. M. Emmelkamp, M. Krijn, A. Hulsbosch, S. De Vries, M. J. Schuemie, and C. A. van der Mast, "Virtual reality treatment versus exposure in vivo: a comparative evaluation in acrophobia," *Behaviour research and therapy*, vol. 40, no. 5, pp. 509–516, 2002.

[8] S. Bouchard, S. Côté, J. St-Jacques, G. Robillard, and P. Renaud, "Effectiveness of virtual reality exposure in the treatment of arachnophobia using 3d games," *Technology and health care*, vol. 14, no. 1, pp. 19–27, 2006.

[9] L. F. Hodges, B. O. Rothbaum, R. Kooper, D. Opdyke, T. Meyer, J. J. de Graaff, J. S. Williford, and M. M. North, "Presence as the defining factor in a vr application: Virtual reality graded exposure in the treatment of acrophobia," 1994.

[10] B. O. Rothbaum, L. Hodges, S. Smith, J. H. Lee, and L. Price, "A controlled study of virtual reality exposure therapy for the fear of flying." *Journal of consulting and Clinical Psychology*, vol. 68, no. 6, p. 1020, 2000.

[11] P. M. Emmelkamp, M. Bruynzeel, L. Drost, and C. A. G. van der Mast, "Virtual reality treatment in acrophobia: a comparison with exposure in vivo," *CyberPsychology & Behavior*, vol. 4, no. 3, pp. 335–339, 2001.

[12] C. Botella, R. M. Baños, C. Perpiñá, H. Villa, M. Alcañiz, and A. Rey, "Virtual reality treatment of claustrophobia: a case report," *Behaviour research and therapy*, vol. 36, no. 2, pp. 239–246, 1998.

[13] P. Anderson, B. O. Rothbaum, and L. F. Hodges, "Virtual reality exposure in the treatment of social anxiety," *Cognitive and Behavioral Practice*, vol. 10, no. 3, pp. 240–247, 2003.

[14] T. Horigome, S. Kurokawa, K. Sawada, S. Kudo, K. Shiga, M. Mimura, and T. Kishimoto, "Virtual reality exposure therapy for social anxiety disorder: a systematic review and meta-analysis," *Psychological Medicine*, vol. 50, no. 15, pp. 2487–2497, 2020.

[15] S. Bouchard, S. Dumoulin, G. Robillard, T. Guitard, E. Klinger, H. Forget, C. Loranger, and F. X. Roucaut, "Virtual reality compared with in vivo exposure in the treatment of social anxiety disorder: a three-arm randomised controlled trial," *The British Journal of Psychiatry*, vol. 210, no. 4, pp. 276–283, 2017.

[16] S. Clift, G. Hancox, I. Morrison, B. Hess, G. Kreutz, and D. Stewart, "Choral singing and psychological wellbeing: Quantitative and qualitative findings from english choirs in a cross-national survey," *Journal of Applied Arts & Health*, vol. 1, no. 1, pp. 19–34, 2010.

[17] G. A. Dingle, E. Williams, J. Jetten, and J. Welch, "Choir singing and creative writing enhance emotion

regulation in adults with chronic mental health conditions," *British Journal of Clinical Psychology*, vol. 56, no. 4, pp. 443–457, 2017.

[18] N. Baghaei, V. Chitale, A. Hlasnik, L. Stemmet, H.-N. Liang, and R. Porter, "Virtual reality for supporting the treatment of depression and anxiety: Scoping review," *JMIR mental health*, vol. 8, no. 9, p. e29681, 2021.

[19] W.-P. Brinkman, C. Van der Mast, G. Sandino, L. T. Gunawan, and P. M. Emmelkamp, "The therapist user interface of a virtual reality exposure therapy system in the treatment of fear of flying," *Interacting with computers*, vol. 22, no. 4, pp. 299–310, 2010.

[20] P. L. Anderson, M. Price, S. M. Edwards, M. A. Obasaju, S. K. Schmertz, E. Zimand, and M. R. Calamaras, "Virtual reality exposure therapy for social anxiety disorder: A randomized controlled trial." *Journal of consulting and clinical psychology*, vol. 81, no. 5, p. 751, 2013.

[21] J. Nielsen, *Usability engineering*. Elsevier, 1994.

[22] T. Bjørner, *Why'Qualitative Methods for Consumer Research'?* Hans Reitzels Forlag, 2015.

[23] M. Ragot, N. Martin, S. Em, N. Pallamin, and J.-M. Diverrez, "Emotion recognition using physiological signals: laboratory vs. wearable sensors," in *International Conference on Applied Human Factors and Ergonomics*. Springer, 2017, pp. 15–22.

FROM MIDI TO RICH TABLATURES: AN AUTOMATIC GENERATIVE SYSTEM INCORPORATING LEAD GUITARISTS' FINGERING AND STYLISTIC CHOICES

Pierluigi BONTEMPI (0000-0002-4360-8708) [1], **Daniele MANERBA** (0000-0002-3502-5289) [2], **Alexandre D'HOOGE** (0000-0003-1634-3406) [3], and **Sergio CANAZZA** (0000-0001-7083-4615) [4]

[1,4] *Centro di Sonologia Computazionale, Department of Information Engineering*, **University of Padua**, Italy
[2] *Department of Information Engineering*, **Università degli Studi di Brescia**, Italy
[3] **Univ. Lille, CNRS, Centrale Lille, UMR 9189 CRIStAL**, F-59000 Lille, France

ABSTRACT

Although the automatic identification of the optimal fingering for the performance of melodies on fretted string instruments has already been addressed (at least partially) in the literature, the specific case regarding lead electric guitar requires a dedicated approach. We propose a system that can generate, from simple MIDI melodies, tablatures enriched by fingerings, articulations, and expressive techniques. The basic fingering is derived by solving a constrained and multi-attribute optimization problem, which derives the best position of the fretting hand, not just the finger used at each moment. Then, by analyzing statistical data from the *mySongBook* corpus, the most common clichés and biomechanical feasibility, articulations, and expressive techniques are introduced. Finally, the obtained output is converted into *MusicXML* format, which allows for easy visualization and use. The quality of the tablatures derived and the high configurability of the proposed approach can have several impacts, in particular in the fields of instrumental teaching, assisted composition and arranging, and computational expressive music performance models.

1. INTRODUCTION

Fretted string instruments (which include guitar, banjo, ukulele, and bass guitar, among others) allow the musician to play most of the notes on at least two different strings, at different frets. In addition, distinct fingers (usually index, middle, ring, or little finger) of the fretting hand can be used to produce the same note, on the same string and at the same fret. In some cases the notes can also be played on the open strings. In music, the term *fingering* is used to indicate which fingers and, consequently, hand positions should be used to perform a part [1]. Opting for well-reasoned fingerings is crucial, either for expressive reasons or to enable the musician to minimize performance effort.

The performer can then make choices, in addition to fingering, regarding articulations and expressive techniques from a range of options generally associated with the specific instrument being played. The term *articulation* refers to how a single note is sounded. Each articulation corresponds to a specific performance technique and generates sounds endowed with distinctive sonic characteristics. The parameters involved may be attack, decay, timbre, dynamics, and in some cases even pitch. The term *expressive technique* instead refers to all those techniques used to imprint variations in sound parameters for specifically expressive purposes. Clearly, there may be at least partial overlap between the two categorizations. Since we specifically address lead electric guitar parts, in this context the most common articulations are those of the plucked string (usually with the use of a plectrum) and the legato obtained through *hammer-on* (a finger of the fretting hand strikes the string from above, causing it to resonate) or *pull-off* (a finger of the fretting hand "plucks/pulls" the string, when it is already vibrating, reaching a note lower than the preceding one). In terms of expressive techniques, we consider *vibrato* (the pitch of the note is made to oscillate with the use of the finger(s) of the fretting hand or the whammy bar), *slide legato* (or glissando), and *bending* (pushing the fingered string upward or pulling it downward, perpendicular to the neck, the pitch of the note is raised) [2].

In this paper, we propose an automatic system able to generate, from monophonic MIDI melodies (including note-on and note-off messages only) [3], electric guitar tablatures enriched by fingering, articulations, and expressive techniques. The basic fingering is derived by solving an optimization problem that considers the possibility of playing the same notes on different string-fret combinations or on open strings as well as biomechanical constraints, and that tries to minimize the time needed for the hand movements and the performer's discomfort, while also accounting for timbre characteristics associated with the generated tablature. The system also keeps track of the position of the fretting hand, not just of the finger used at each moment. Then, taking into account the most common clichés and biomechanical feasibility, articulations, and expressive techniques are introduced. The usage frequencies of such features are derived by analyzing statistical data from the *mySongBook* corpus, an impressive proprietary collection of guitar-oriented transcriptions, already used in the past in other publications [4, 5]. Finally, for ease of visualization

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

and use, the result is converted into *MusicXML* format.

Two important observations are in order. First, we address by choice guitar solos and melodies composed of single notes only and leave the possibility of playing multiple notes simultaneously for future system evolutions. However, it should be noted that lead guitar sections include a very high percentage (>90%) of single notes (see Cournut et al. 2021 [4]). Second, our system is based on symbolic encodings of music, i.e., MIDI, a specially developed dataframe, and MusicXML. Clearly, the automatic generation of transcriptions from audio recordings of guitar performances, a subject of significant attention in the literature (see, among other works, [6, 7]), shares the same object of study, but observes it backward, i.e., from performance to tablature or score. In our case, the goal is to move from a minimalistic score (represented in MIDI format) to a guitar tablature enriched with fingering, articulations, and expressive techniques.

The contribution of this work is multifaceted. On the one hand, earlier approaches to automatic fingering generation based on optimization techniques are somehow revived and greatly expanded (see Section 2). To the best of our knowledge, none of the approaches for automatic fingering generation presented in the past considered simultaneously all of the parameters and attributes described later, nor allowed for a level of user-side customization comparable to that proposed here. On the other hand, the automatic insertion of articulations and expressive techniques is deeply explored, while being a largely marginal subject of research in the past (a notable exception is the work by Sarmento et al. 2023 [8], although focused on automatic music generation). We also believe that this work can constitute an important piece toward the construction of models for computational musical expressiveness, in teaching, composition, assisted arranging of guitar parts, and, indirectly, in the stylistic analysis of guitarists. Lastly, the *mySongBook* corpus confirms itself to be a precious resource for understanding contemporary guitar practice and for the generation of automatic systems.

The rest of the paper is organized as follows. Section 2 presents the optimization approach followed to derive the basic fingering, while Section 3 presents first the *mySongBook* corpus and the statistical data derived from it, and then describes the rule system developed for the automatic addition of articulations and expressive techniques. In Section 4, the generation of the MusicXML code is introduced and an example lead guitar part processed by the system is presented, in the form of *rich* tablature and score in standard notation. Finally, we conclude in Section 5, also discussing some possible applications and future developments.

2. FROM A MIDI MELODY TO THE BASIC FINGERING

The automatic search for valid or optimal fingerings in relation to specific musical instruments has a scientific literature of significant size that cannot be entirely reviewed here. Hence, we first report the main existing approaches related to plucked string instruments, highlighting the gaps we are trying to fill. Then, we describe our novel and comprehensive methodology.

2.1 Existing approaches for fingering optimization

Sayegh (1989) [9] tackles the fingering procedure by solving a Shortest Path Problem (SPP) on a weighted and layered directed graph (a so-called *Viterbi* network) in which each layer corresponds to each note of the sequence and contains nodes corresponding to admissible fingerings of such a note. Since only position and string changes between a note and the consecutive one affect the fingering attributes, the weights of such attributes can be directly associated with arcs between two consecutive layers. Concerning the considered attributes, two broad classes of rules may be identified. The first one relates to ease of execution and the second to the homogeneity of the sound generated.

A similar approach, albeit slightly more refined, is proposed in Radicioni et al. (2004) [10]. Here, not only movements along and across the fretboard are taken into consideration, but also finger span, and minimum/maximum spans between each pair of fingers are used to evaluate the level of difficulty of the transitions between notes and identify when hand repositioning is needed. The sequence is segmented into musical phrases at the preprocessing phase and the optimization takes place first of all within the individual phrases, then a valid connection between phrases is created. The approach of Miura et al. (2004) [11] is also based on the segmentation of the melody while the objective is to minimize the number of movements of the fretting hand wrist (along the neck) within each section. The trade-off emerges between the increase of finger span and the hand position changes, where the latter is however considered less uncomfortable for a beginner. Interestingly enough, techniques such as hammer-on, pull-off, and slide are considered.

Other published works focus more on algorithmic aspects of the fingering. In particular, Tuohy and Potter (2005) [12] propose a Genetic Algorithm for creating polyphonic guitar tablatures. Their fitness function is based on hand movements along and across the neck, and fretting hand configurations. However, the system does not include information about which finger should be used to play each note. A Hidden Markov Model (HMM) is used in Hori et al. (2013) [13] to derive the basic fingering of monophonic and polyphonic guitar parts. The fretboard hand arrangements are considered as the hidden states, and the note sequence as the observed one. In a subsequent work by Hori and Sagayama (2016) [14], the authors try to simplify the most difficult movements required to play a part, instead of aiming at minimizing the difficulty of the overall execution. The paper introduces a variant of the basic algorithm for multi-layered networks, called the *minimax Viterbi algorithm*, to solve the problem. When applied to an HMM-based fingering decision network, the algorithm looks for the sequence of the hidden states that maximizes the minimum transition probability on the sequence (transition probabilities are large for easy moves and small for difficult ones). The reader is referred

to Roohnavazfar et al. (2019) [15] for more recent and sophisticated approaches for finding optimal paths in general multi-stage decision networks.

Finally, a Dynamic Programming approach is used in Raboanary et al. (2017) [16] to minimize performance effort on monophonic bass guitar lines, with special emphasis on limiting movements along the neck.

2.2 A new comprehensive optimization framework

Our approach for generating the basic fingering consists in the formalization, development, and resolution of an optimization problem, which generalizes all those already presented in the literature that appeared under the simplified form of an optimal path search through a multi-layered network (see, e.g., Sayegh 1989 [9]). In particular, we enrich the quality of the fingering by introducing i) additional requirements, that affect the available feasible fingerings for each note of the melody, and ii) simultaneous attributes to optimize, that affect the cost of the transition from the fingering of a note to that of the following one.

Concerning the attributes of the fingering to be optimized, we identified three macro-categories:

- *time-related* attributes: relate to movements of the fretting hand along (*position change*, PC) and across (*string change*, SC) the neck. Their weights depend on the time needed to physically reach the fingering by the hand and the fingers.
- *discomfort-related* attributes: relate to finger span (*hand spread*, HS) and to playing notes on specific parts of the fretboard (e.g., very near or very far from the headstock). Their weights depend on the physical discomfort generated by the fingering in the player.
- *timbre-related* attributes: relate to attributes that affect the type of sound output produced. Their weights depend on the sound quality generated by the fingering, according to the player's intent and taste. In this regard, we consider here the frequency of use of open strings.

We remark that, differently from most of the existing approaches, the system keeps track of the fretting hand position (i.e., the fret at which the index finger is located at each moment) to estimate the spreads between fingers and the longitudinal hand movements.

Moreover, our optimization setting provides the user with great flexibility, allowing the customization of several aspects and preferences. In particular, the system can be finely configured about:

- the intrinsic characteristics of the instrument (number of strings and frets, and notes generated by each open string or string/fret combination);
- the fretboard area on which it is more comfortable to play. Typically the preferred area is the middle of the neck, between the fifth and twelfth positions, as noted by Cournut et al. (2021) [4] and by Heijink et al. (2002) [17] concerning classical guitar;
- the maximum and the minimum allowed spread between each finger and the index finger. Constantly keeping track of the position of the index finger allows for a potentially more accurate assessment of the hand spread than the approach traditionally followed, which only monitors the spread between the finger used to play one note and the one used for the immediately following one. It also makes it possible to correctly assess position changes, i.e., shifts of the hand along the neck (see below);
- the time required for the hand to make longitudinal movements (proportional to the number of frets to be covered) and vertical movements (proportional to the number of strings covered). Transitions for which the *Inter Onset Interval* (IOI) is not sufficient are discarded from the possible alternatives;
- how fingers are used. For example, it could be denied to use the same finger to play two contiguous notes unless they are played at the same fret (as by a *barrè* o *semi-barrè* technique) or on the same string (so to allow sliding techniques);
- the proportional level of penalty to be associated with each attribute to be optimized, e.g., for position changes, string changes, use of open strings, finger spread, and distance from the optimal fretboard area.

2.3 Implementation details

In the following, we briefly describe the system implementation from a purely technical point of view.

Given a monophonic melody in MIDI format, a dataframe is generated in Python language using the *Mido* library (`https://github.com/mido/mido`, last accessed: 2024/03/07). For each note, the dataframe contains pitch, start and end positions in time, duration, IOI, and all the data needed later to generate the final MusicXML file. Data on the structure of the instrument (number of strings and frets, and notes generated by each open string and string/key combination) and the configuration parameters of our optimization problem are also generated.

The obtained dataframe is then processed through the *IBM ILOG CPLEX Optimization Studio* software, v20.1.0 (`https://www.ibm.com/products/ilog-cplex-optimization-studio`, last accessed: 2024/03/07), in which we implemented our optimal fingering problem as a static and compact Mathematical Programming formulation. The problem is described via the CPLEX's dedicated *Optimization Programming Language*. The software returns the string/fret combination (and the associated finger to be used to press the string) or the open string on which to play each note implying the best achievement in terms of time-, discomfort-, and timbre-related attributes.

Figures 1 and 2 show the same melody processed by the optimizer with different attributes' importance.

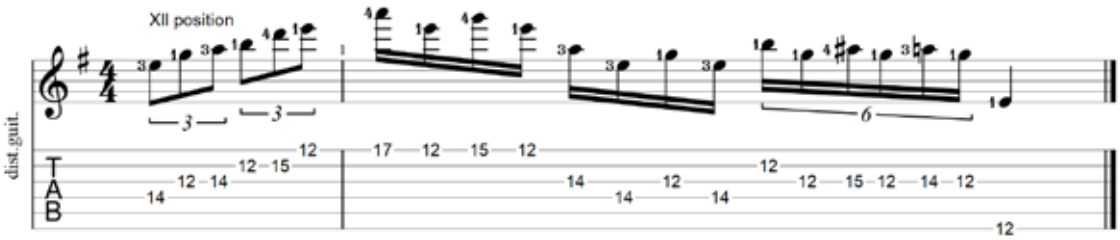


Figure 1. To obtain this solution, the least penalized feature was the string change (SC). The hand spread (HS) was penalized twice as much as SC, while the position change (PC) was penalized twice as much as HS.

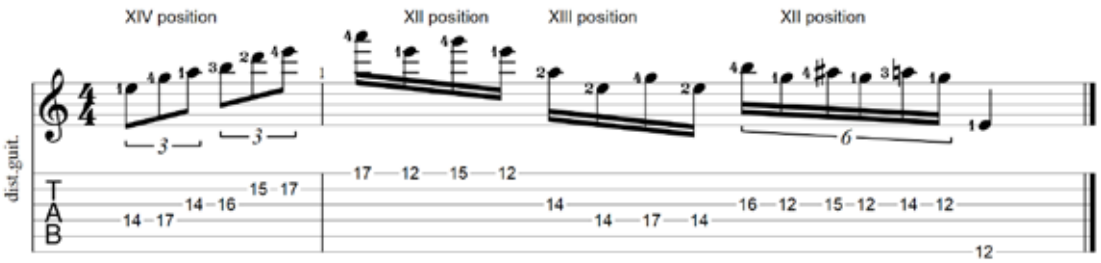


Figure 2. To obtain this solution, the least penalized feature was HS. PC was penalized twice as much as HS. SC was penalized twice as much as PC.

The numbers placed next to the heads of the notes in standard notation refer to the finger of the fretting hand to be used, i.e., 1-index, 2-middle, 3-ring, and 4-little finger. Both results are biomechanically valid, but significantly different from each other, reflecting the high configurability of the approach.

3. FROM THE BASIC FINGERING TOWARD A *RICH* TABLATURE

In this Section, we describe how the dataframe is enriched with articulations and expressive techniques. First, we extracted the distributions of the main techniques from the *mySongBook* dataset to use them as statistical references. Then, we use a rule-based system for the actual insertion of the techniques, allowing wide freedom of configuration to the user.

3.1 Statistical data extraction

The *mySongBook* corpus comprises more than 2000 scores/tablatures transcribed by professional musicians (`https://www.guitar-pro.com/c/18-music-scores-tabs`, last accessed: 2024/03/07), of different genres. The transcriptions are made with *Guitar Pro* software, produced by *Arobas Music*, and are geared first and foremost to an audience of guitarists. However, they usually also contain vocal parts or parts of other instruments, such as bass guitar or drums. Given the object of investigation of this paper, the statistical data described below were extracted from the sections labeled as *lead electric guitar parts*. These parts are extracted after eliminating rhythmic guitar sections, identified using the procedure described in Régnier et al. (2021) [18]. Then, the quantity of playing techniques on each string is determined and divided by the total number of notes per string to obtain the proportion of notes with each technique. The obtained ratios are illustrated in Figure 3.

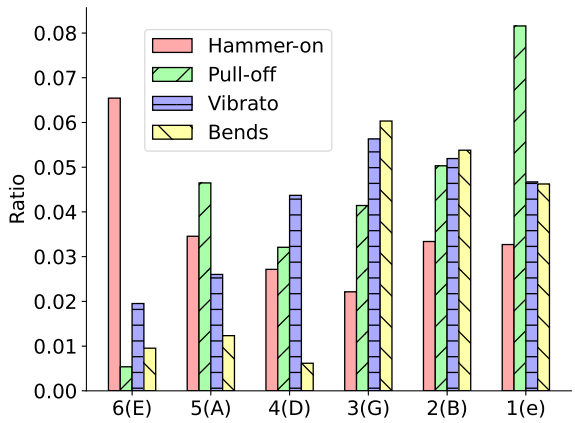


Figure 3. Ratio of the use of the considered techniques to the total number of notes for each string. Strings are labeled according to guitar standard tuning, the 6-th string being the low “E” and the 1-st string the high “e”.

It can be observed that vibrato and bends are used similarly on higher strings. However, bends are rare on lower strings because thicker strings are harder to bend, while a vibrato is a smaller movement that is manageable even on stiffer strings. It should also be noted that, even if hammer-on and pull-off are techniques theoretically similar, their distribution across strings is quite different.

Our system uses the statistical data presented for deciding on the insertion of articulations and expressive techniques, as described in Section 3.2.

The data were also used as a comparison benchmark for the basic fingerings generated by the optimization procedure. On a test dataset made up of lead parts comprising a total of around 1500 notes and by setting the configuration weights in such a way as to balance the different discomfort factors, it was observed that the distribution of notes for each string was reasonably similar to that found in the entire *mySongBook* corpus (Figure 4).

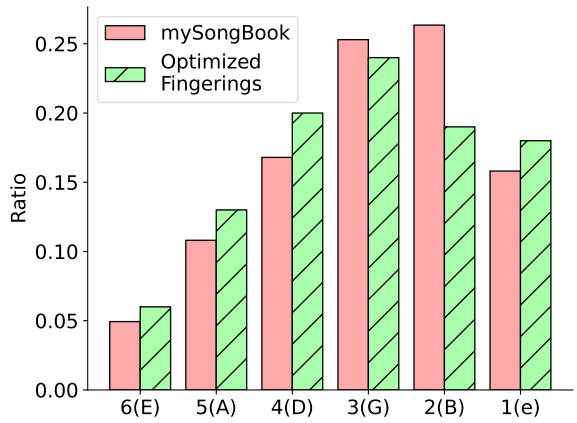


Figure 4. Ratio of the notes played on the different strings to the total number of notes in the test dataset (applying our optimized fingering) and the entire *mySongBook* corpus.

Finally, specifically regarding bending techniques, a manual analysis of numerous transcriptions taken from the *mySongBook* corpus and from teaching manuals [19–21] revealed some typical situations where bends are used, specifically in the presence of two or more consecutive notes in unison, or of groups of three consecutive notes with first and third notes equal to each other and the second one with higher or lower pitch within the limits consonant with the technique in question (e.g., G4-A4-G4, where A4 is performed in bending, keeping the fret/chord/finger combination steady), or when two contiguous notes can be played at the same fret of the same string, using a bend on one of them. Such typical situations are again used as a rule-of-thumb for bending decisions, as described in the next Section.

3.2 Inclusion of articulations and expressive techniques

The dataframe generated from the MIDI melody, combined with the fingering solution obtained from the optimization procedure, is further processed in Python to insert articulations and expressive techniques. Those considered are listed below, along with a summary of the rules for evaluating the possibility or opportunity of their execution:

- hammer-on: only the parameter of biomechanical feasibility is evaluated. If the note is preceded by another lower-pitched note on an open string or played with a lower finger on the same string, the articulation is possible;
- pull-off: only the parameter of biomechanical feasibility is evaluated. If the note is preceded by another higher-pitched note played with a higher finger on the same string, the articulation is possible. The pull-off is always possible between fretted and open strings;
- vibrato: two user-customizable duration values are set; if the note duration reaches the first, the expressive technique is considered possible; if it reaches the second it is estimated highly likely. Vibrato is not allowed on open strings;
- slide legato: only the parameter of biomechanical feasibility is evaluated. If a note is followed by another note on the same string and at a different fret, played with the same fretting hand finger, the articulation is possible;
- bending: the field of bending is highly articulated and complex to model [5]. For this first implementation of the software, only basic upward bends (and possibly the releases associated with them, if the following note is played with the same finger and at the same string/fret combination) were considered. The user is first asked to set the maximum amplitude in semitones of the bends that can be performed by each finger of the fretting hand. Setting this parameter to 0 excludes fingers (typically the little finger) from the possibility of doing bends. Bending is considered possible if the hypothetical application note has a duration equal to or greater than a user-settable value. Biomechanical compatibility between the execution of the bending and the fingering of the previous and next notes is also taken into consideration. The procedure tries to include bends by recreating where possible the cliché situations described in Section 3.1.

The actual distribution of articulations or expressive techniques depends on the user's choices and the reference statistics described in Section 3.1. The user can let such statistics be taken as targets to be achieved, consistent with the biomechanical possibility of the execution. In other words, the procedure tries to include, where possible according to the conditions described above, the techniques to respect the proportions between technique use and the total number of notes for each string. If the points of possible insertion of the techniques are insufficient in number to reach the targets, the procedure will simply insert the techniques wherever possible, without forcing their insertion into unnatural contexts. If, on the other hand, the potential insertion points are more numerous than the target, the use of the different techniques is chosen on a random basis, to get as close as possible in the overall part to the target values. In the case of vibrato, priority is given to the longest notes.

The user also has the possibility to customize the reference targets in relation to each of the above-mentioned techniques, so as to be able to better adapt the system's response according to musical genre, personal preferences, or educational needs. The user can also ask the procedure to insert vibratos, hammer-ons, and pull-offs wherever possible. Only in the case of the slide legato, given the small number of potential occurrences, it was deemed appropriate to limit ourselves to asking the user if s/he would like different notes played with the same finger on the same string to be connected with this technique.

We remark that, clearly, the system also takes into consideration the compatibility between the different techniques, so as not to overlap conflicting ones.

4. MUSICXML CONVERSION AND GRAPHICAL RENDERING

Once the dataframe is completed, including all useful information concerning pitch, timing, use of open or fingered strings, articulations, and expressive techniques, a file in MusicXML format is generated via a dedicated Python script. The MusicXML format is considered a standard for the interchange and distribution of musical scores, provides for tablature encoding, and is supported by many music notation programs [22, 23]. We opted for *Guitar Pro 8*, to obtain a graphical rendering of the processed parts.

The final result concerning a sample lead part, processed by the entire system, is shown in Figure 5. The melody is an original work composed by the first author, but any MIDI melody could be used. The part was specifically written by imagining a standard 12-bar blues harmonic progression in A. Here, the reader may notice that the fingering is biomechanically valid and comfortable. The rich presence of bends, vibratos, hammer-ons, and pull-offs is because higher target values were given when inserting articulations and expressive techniques than those found in the *mySongBook* corpus, to better embody the blues or rock/blues feel of the phrasing [21]. A slide legato also appears in the last bar.

5. CONCLUSIONS

We presented an innovative automatic system for generating (starting from simple MIDI melodies) lead guitar tablatures, enriched with fingering as well as articulation and expressive techniques. For the basic fingering, we used an optimization approach significantly richer than the ones already existing in the literature, that allowed us to embed more complex requirements. Moreover, we also developed a procedure that provides a very high level of customization for the user, both in terms of biomechanical preferences/requirements and stylistic intents. Starting from the basic fingering, a user-customizable rule-based system, which evaluates note by note the biomechanical feasibility and probability of use of specific articulations and expressive techniques, introduces such techniques whenever appropriate. Finally, the generated dataframe is converted into MusicXML format, compatible with numerous music notation software. The code is available upon request to the authors.

The computational generation of optimal or alternative *rich* fingerings for a given melodic part can be of interest, beyond the support of the performer in the study phase, in multiple areas such as teaching [24], automatic music generation [25, 26], computational performative expressiveness [27, 28], stylistic analysis, evaluation of existing tablatures [29], and testing of audio-to-tablature models. Both the Centro di Sonologia Computazionale of the University of Padua [30] and the TABASCO project team (`http://algomus.fr/tabasco/`, last accessed: 2024/05/16), to which part of the authors belong, share previous relevant work and scientific interest in relation to these issues. We believe that in each of the fields mentioned above the verification of performative plausibility and the possible generation of alternative *rich* fingerings can prove valuable (if not indispensable) aids.

The future improvements of the system's functionalities may comprise the management of lead parts including multiple simultaneous notes, the possibility of inserting decisions concerning articulations and expressive techniques within the optimization procedure, as well as the application of the system in different fields such as teaching (even self-taught) and computational musical expressiveness. Moreover, the much-needed integration of a system such as the one proposed here into the workflow of contemporary music production, often based on the use of virtual instruments, would allow even those who are not expert guitarists to obtain more realistic and convincing parts. Finally, it should be noted how a system such as the one proposed cannot be easily validated, as a given starting melody may correspond to many divergent solutions, equally valid but responding to different expressive intentions and personal preferences. In other words, it would not be correct to compare the output of the system to specific actual performances or existing transcriptions to sanction or not its validity. Although the results obtained so far have been evaluated by the authors as fully satisfactory and realistic, it would therefore be desirable in the future to subject them to the judgment of a wider audience of experienced guitarists.

Acknowledgments

This work is partially supported by the French National Research Agency, in the framework of the project TABASCO (ANR-22-CE38-0001). The authors would also like to thank *Arobas Music* for providing the dataset described in Section 3.1.

6. REFERENCES

[1] M. Lindley, G. Jenkins, R. Slatford, P. Walls, S. Monosoff, A. Crum, S. Wijsman, A. Myers, M. Ecochard, and B. Haynes, "Fingering," in *Grove Music Online*. Oxford University Press, 2001.

[2] D. R. Grimes, "String theory - the physics of string-bending and other electric guitar techniques," *PLOS ONE*, vol. 9, no. 7, pp. 1–9, 07 2014.

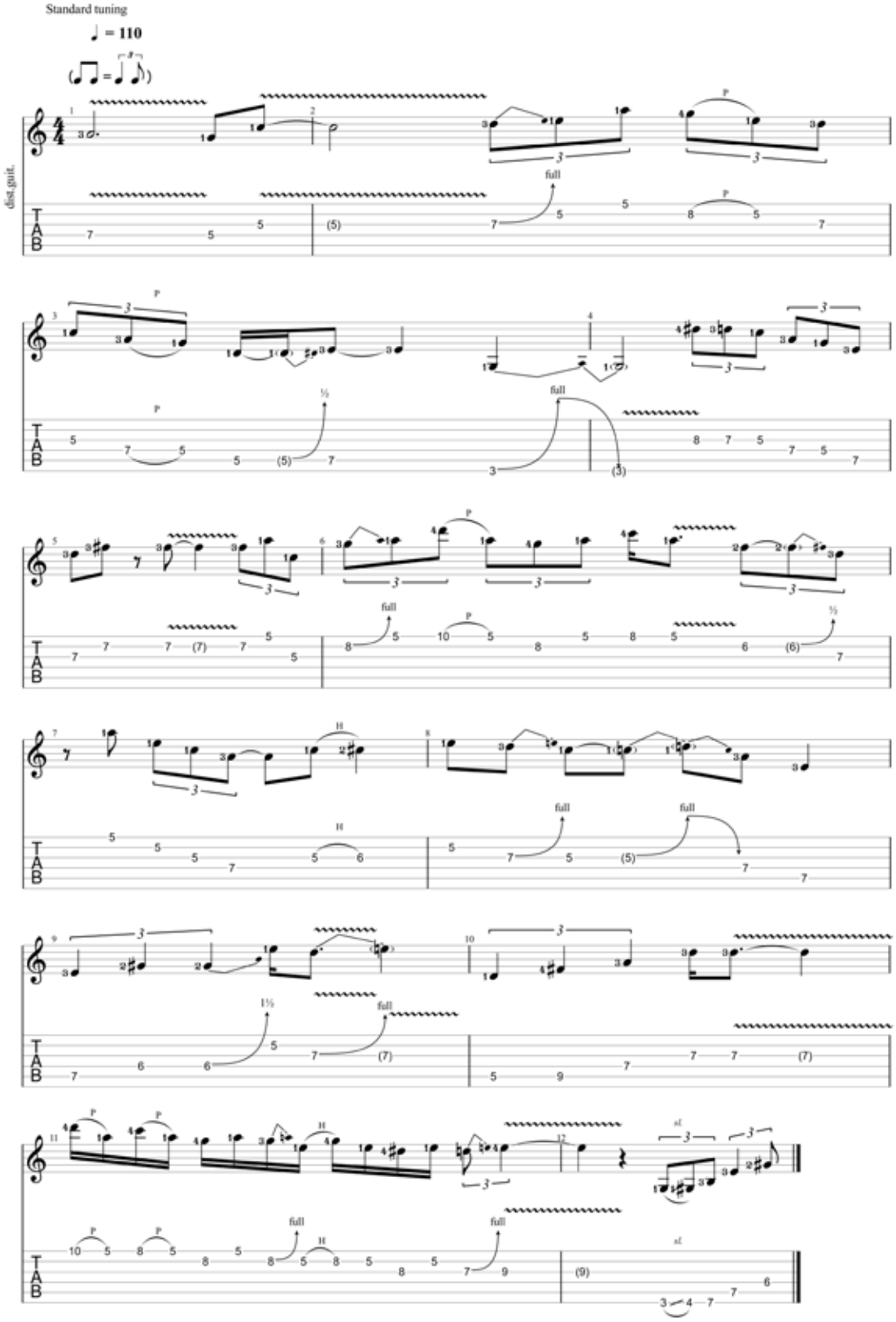


Figure 5. Rich tablature and standard notation of a sample lead part automatically generated by our system.

[3] B. Moog, "Midi," *Journal of Audio Engineering Society*, vol. 34, no. 5, 1986.

[4] J. Cournut, L. Bigo, M. Giraud, N. Martin, and D. Régnier, "What are the most used guitar positions?" in *Proceedings of the International Conference on Digital Libraries for Musicology (DLfM)*, July 2021, pp. 84–92.

[5] A. D'Hooge, L. Bigo, and K. Déguernel, "Modeling

bends in popular music guitar tablatures," in *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference*, November 2023.

[6] J. Michelson, R. Stern, and T. Sullivan, "Automatic guitar tablature transcription from audio using inharmonicity regression and bayesian classification," in *Audio Engineering Society Convention 145*. Audio Engineering Society, 2018.

[7] A. Wiggins and Y. E. Kim, "Guitar tablature estimation with a convolutional neural network." in *ISMIR*, 2019, pp. 284–291.

[8] P. Sarmento, A. Kumar, D. Xie, C. Carr, Z. Zukowski, and M. Barthet, "Shredgp: Guitarist style-conditioned tablature generation with transformers," in *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*. Zenodo, 2023.

[9] S. I. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[10] D. Radicioni, L. Anselma, V. Lombardo *et al.*, "A segmentation-based prototype to compute string instruments fingering," in *Proceedings of the Conference on Interdisciplinary Musicology*, vol. 17, 2004, p. 97.

[11] M. Miura, I. Hirota, N. Hama, and M. Yanagida, "Constructing a system for finger-position determination and tablature generation for playing melodies on guitars," *Systems and Computers in Japan*, vol. 35, no. 6, pp. 10–19, 2004.

[12] D. Tuohy and W. Potter, "A genetic algorithm for the automatic generation of playable guitar tablature," in *Proceedings of the International Computer Music Conference (ICMC)*, 01 2005, pp. 499–502.

[13] G. Hori, H. Kameoka, and S. Sagayama, "Input-output hmm applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, no. 2, pp. 264–271, 2013.

[14] G. Hori and S. Sagayama, "Minimax viterbi algorithm for hmm-based guitar fingering decision." in *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference*, 2016, pp. 448–453.

[15] M. Roohnavazfar, D. Manerba, J. C. De Martin, and R. Tadei, "Optimal paths in multi-stage stochastic decision networks," *Operations Research Perspectives*, vol. 6, p. 100124, 2019.

[16] T. H. Raboanary, F. H. Randriamahenintsoa, H. A. Raboanary, T. M. Raboanary, and J. A. Raboanary, "Finding optimal bass guitar fingerings," in *Proceedings of the IEEE Africon Conference*. IEEE, September 2017, pp. 65–71.

[17] H. Heijink and R. G. Meulenbroek, "On the complexity of classical guitar playing: functional adaptations to task constraints," *Journal of Motor Behavior*, vol. 34, no. 4, pp. 339–351, 2002.

[18] D. Régnier, N. Martin, and L. Bigo, "Identification of rhythm guitar sections in symbolic tablatures," in *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference*, November 2021.

[19] P. Fischer, *Masters of Rock Guitar*. Ama Verlag, 2000.

[20] T. Stetina, *Metal lead guitar*. Hal Leonard, 1992.

[21] J. Wheatcroft, *Improvising Blues Guitar*. Schott & Co. Ltd, 2008.

[22] M. Good, "Musicxml: An internet-friendly format for sheet music," in *Proceedings of XML conference and exposition*, December 2001, pp. 03–04.

[23] M. Good and G. Actor, "Using musicxml for file interchange," in *Proceedings of the International Conference on WEB Delivering of Music (WEDELMUSIC)*. IEEE, 2003, p. 153.

[24] T. Izumi, "Cognitively oriented design of a multimedia system to learn guitar fingering," in *Proceedings of the International Conference on Computers in Education (ICCE)*, vol. 1, 2002, pp. 789–791.

[25] F. Carnovalini and A. Rodà, "Computational creativity and music generation systems: An introduction to the state of the art," *Frontiers in Artificial Intelligence*, vol. 3, 2020.

[26] M. Civit, J. Civit-Masot, F. Cuadrado, and M. J. Escalona, "A systematic review of artificial intelligence-based music generation: Scope, applications, and future trends," *Expert Systems with Applications*, vol. 209, p. 118190, 2022.

[27] C. E. Cancino-Chacón, M. Grachten, W. Goebl, and G. Widmer, "Computational models of expressive music performance: A comprehensive and critical review," *Frontiers in Digital Humanities*, vol. 5, 2018.

[28] P. Bontempi, S. Canazza, F. Carnovalini, and A. Rodà, "Research in computational expressive music performance and popular music production: A potential field of application?" *Multimodal Technologies and Interaction*, vol. 7, no. 2, 2023.

[29] A. Radisavljevic and P. F. Driessen, "Path difference learning for guitar fingering problem," in *Proceedings of the International Computer Music Conference (ICMC)*, November 2004.

[30] S. Canazza and G. De Poli, "Four decades of music research, creation, and education at Padua's Centro di Sonologia Computazionale," *Computer Music Journal*, vol. 43, no. 4, pp. 58–80, 2020.

I GOT RHYTHM, SO FOLLOW ME MORE: MODELING SCORE-DEPENDENT TIMING SYNCHRONIZATION IN A PIANO DUET

Akira MAEZAWA (akira.maezawa@music.yamaha.com) [1]

[1] **Yamaha Corporation**, Hamamatsu, Japan

ABSTRACT

Automatic accompaniment systems synchronizing software playback with human musicians require a natural, easy-to-manipulate music synchronization model. However, existing methods cannot synchronize in a way aware of the musical context. This paper presents an interpretable model of the sensorimotor synchronization (SMS) model based on the user's music score. The model maps the music score to an easy-to-interpret linear model of phase and period correction using a deep neural network, allowing timing synchronization to be aware of the music score. The evaluation shows the proposed method achieves lower estimation error than a model that is unaware of the music score or one that is linearly dependent on the score.

1. INTRODUCTION

Automatic accompaniment is a technique that enables a machine to accompany a human musician's performance by synchronizing its playback to the human performer. It helps accompany users who cannot play in time or wish to express themselves through artistic timing fluctuations. For the machine to provide comfortable accompaniment for humans, the machine should follow the musician and synchronize naturally, as a human accompanist would do. This is challenging because playing in a musical ensemble is an intricate art: multiple musicians interact with each other in a way that is dependent on the musical context, perception, and performance. In practice, it is also a challenge to provide an expressive model whose behavior is easy to analyze and manipulate.

Existing accompaniment systems lacked a simple yet expressive accompaniment playback model that is aware of musical contexts. Earlier methods required the composer to specify how smoothly it should play an accompaniment snippet in response to the user's performance [1]. Some methods have used heuristics of performance of music score to adjust simple parameters for accompaniment playback [2, 3], but such a heuristic approach can potentially miss intricate factors in timing synchronization. Many models use rehearsals to learn a synchronization model [4, 5] but cannot generalize outside the rehearsed pieces. A recent method has arrived at a simple and interpretable model inspired by sensorimotor synchronization (SMS) [6], but it has fixed parameters or the response throughout the piece, unaware of the musical contexts.

This paper proposes a method for timing synchronization that takes into account musical contexts, can generalize across different pieces, and is simple to interpret and adjust. Inspired by models of SMS [7] that are simple and robust, we present an extension of such a model that is conditioned on a music score. The proposed SMS model is aware of the music score thanks to the conditioning on the score, which allows generalization through training. It is also interpretable and allows intuitive manual modifications thanks to the simple formulation of the SMS model.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

2.1 Automatic accompaniment

In automatic accompaniment systems [1,3,4,6,8], the playback of the machines is synchronized to the human player. A model of feasible playback time sequence is assumed, and its parameters are learned through rehearsals. For example, a dynamic Bayesian network describing the timing of a performance of a known piece of music [5] or a linear dynamical system using score and performance features [4] can be used to train how each performer plays a given piece of music. Typically, these are confined to a given piece of music.

To generalize synchronization to new pieces, designing a general model of timing synchronization is necessary. A line of work uses performance features such as IOI stability to compute who leads an ensemble [2]. Another line of work uses the music score features such as note densities [3] for timing synchronization. These use features that have been heuristically designed. To circumvent the manual design of hand-crafted features, our previous work trained a deep neural network to predict the tempo curve given the current performance and the music score [9]. In this paper we apply this approach for timing coordination.

2.2 Music performance rendering

A related task is music performance rendering, which generates expressive parameters such as the velocity and onset timing deviations given a music score. Here, deep neural networks have been successful and modeling the mapping between the score and the performance [10–13]. We use deep neural networks to learn the sensorimotor synchronization parameters, which can be used to control machine playback in a human music ensemble.

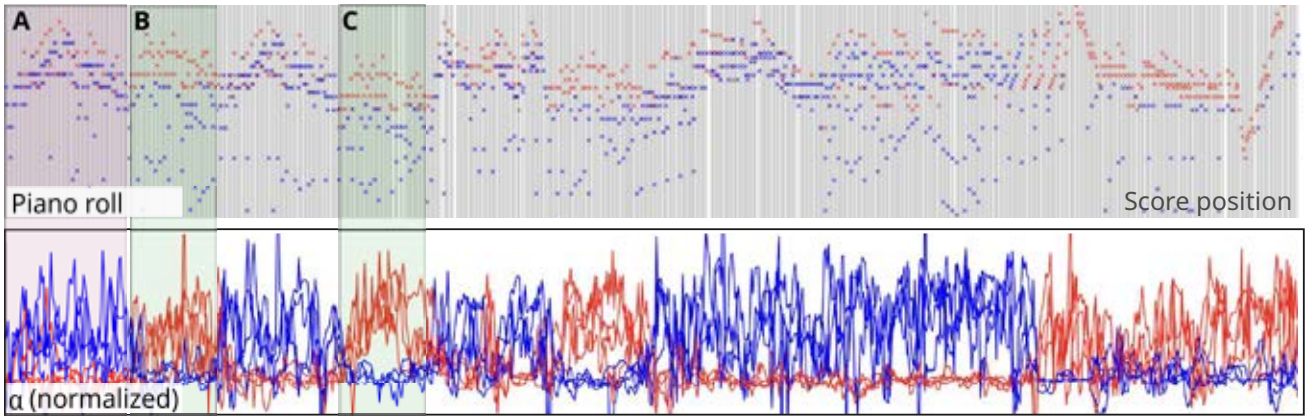


Figure 1. An example of normalized linear phase correction parameter α over four takes and the corresponding piano-roll (piano duet arrangement of Mendelssohn's "Song without words"). The red line represents piano 1, and the blue line represents piano 2. The roles of the two parts are interchanged in segments "A" and "B." Segments "B" and "C" are modulations of each other.

2.3 Sensorimotor Synchronization

Sensorimotor synchronization (SMS) is the phenomenon of human movement coordination in response to an external rhythm, which depends on various factors [14]. For computational modeling of SMS, many studies use linear models [7, 15, 16]. These explain period and phase work in terms of the asynchrony between humans and external stimuli, typically via low-order models like ARMA(1,2) under a restricted parameter space. Some studies implement such cognition-inspired models for SMS for automatic accompaniment systems [2, 6, 17]. Whereas these models assume that SMS model parameters are independent of the musical context, given that SMS depends on the rhythm [18], we hypothesize that the model parameters are better expressed by conditioning on the music score.

3. OUR METHOD

In our previous research [9], we have employed deep learning to extract features for estimating the tempo curve governed by a linear model. It offered the flexibility of a deep neural network while also providing an interpretable system. In this paper, we extend this idea to learning synchronization as well.

3.1 Model of synchronization

We introduce the synchronization model used in this paper and briefly explore the validity of assuming its parameters depend on the score.

3.1.1 Linear phase/period correction

We first consider the period correction model [16], which extends the linear phase correction model to handle tempo changes. It considers a case where a user is given beat stimuli to tap with an interval of S_k seconds for the kth stimulus, and the user would tap along with an error of A_k seconds. The model considers asynchrony to arise from (1) the motor noise $M_k \sim \mathcal{N}(0, \sigma_M^2)$, (2) the discrepancy between S_k and the expected beat period T_k, a probabilistic value with an underlying beat period t_k, $T_k \sim \mathcal{N}(t_k, \sigma_T^2)$, and (3) a corrective process based on recent asynchronies, that corrects beat phase and period according to parameters α and β, respectively:

$$A_{k+1} = (1-\alpha)A_k + M_{k+1} - M_k + T_k - S_k \quad (1)$$

$$t_k = t_{k-1} - \beta A_k. \quad (2)$$

For this study, we modify the original formulation to allow the inter-stimulus interval S_k and the corresponding T_k to scale by the notated note values. We thus describe the asynchrony also in terms of the notated note values d_k, as follows:

$$A_{k+1} = (1-\alpha)A_k + M_{k+1} - M_k + d_k(T_k - S_k). \quad (3)$$

The inter-stimulus interval is clear in tapping literature, but it is not so clear for automatic accompaniment, where one part may be silent. As an approximation, we take the scheduled playback times of the sequencer when using an automatic accompaniment system or the smoothed playback position of a recording when analyzing a human duet for training.

3.1.2 Does synchronization depend on musical context?

To demonstrate the effect the music score has on the linear synchronization parameters, we plot the period correction parameter α against four takes of an identical piano duet in Figure 1, played by one pair of advanced pianists. The figure shows that the region with α taking on more significant values is consistent with the musical role as implied by the music score (by comparing (A) and (B)) and is consistent with musically similar parts (by comparing (B) and (C)). It thus suggests that the music score is useful for inferring how performers synchronize with each other.

3.2 Parameter prediction model

We consider two models for predicting the parameters α, β, σ_M, σ_T. In both cases, the model takes as the input a two-channel binary piano-roll, $X(c, t, p) \in$

$\{-0.5, 0.5\}^{C \times T \times P}$, where c is the channel index, t is the frame index, p is the pitch index. Given a music score, the human part is assigned to $c = 0$ and the machine part to $c = 1$. The frame index t is computed by the number of 32nd notes elapsed since the beginning, and the pitch index p is computed by taking the pitch of each note event and taking the floor after dividing it by 8, chosen for mathematical convenience, with $P = 16$. The feature thus represents a rough indication of the timing and the register of the current note events.

First, we consider a *linear regression model*, which estimates the parameters by multiple regression that takes the piano roll centered about the current position, with a radius of two beats. For the kth onset time t_k, we compute X_k as a slice of $X(c, t, p)$ centered about t_k with a window of two beats, *i.e.*, from $t = t_k - 16$ to $t_k + 16$. In other words, for the input feature X_k for kth note onset time, we estimate the parameters as follows:

$$[\alpha_k, \beta_k, \sigma_{M,k}, \sigma_{T,k}]^T = W \mathrm{vec}(X_k) + b. \quad (4)$$

where $W \in \mathbb{R}^{4 \times \mathrm{length}(\mathrm{vec}(X_k))}$ and $b \in \mathbb{R}^4$.

Second, we consider a *deep neural network* (DNN) for parameter estimation. It also accepts the piano roll X centered about the current position with a radius of two beats. For the kth onset in the score, we split X_k into X_k^{pre} and X_k^{post}, which are the X_k computed before and after the onset frame t_k. The two features X_k^{pre} and X_k^{post}, respectively, then undergo four layers of a series of convolutional networks with a kernel size of 2 with a dilation factor of 2^l, where l is the zero-indexed layer index, followed by batch normalization, exponential linear unit (ELU) activation and dropout with dropout probability of 0.05. The number of output channels of each layer is $[128, 256, 256, 256]$. The output is average pooled over the frame and pitch axes to yield a pair of 256-dimensional vectors c_k^{pre} and c_k^{post}. These are concatenated with the piano roll evaluated exactly at the kth onset frame, $X(:, t_k, :)$, to arrive at a context vector c_k. It then undergoes three layers of linear layers with output channels of 1000, 200, and 4 with dropout and ELU nonlinearity, except the last layer, to arrive at $[\alpha, \beta, \sigma_M, \sigma_T]$.

4. EVALUATION

4.1 Dataset

We have acquired a dataset of piano duets played by an advanced-level piano duet group. Four-hands arrangements were obtained for the fourteen pieces listed in Table 1, published by Print Gakufu [1]. For each piece, the pianists had time to study the score beforehand and rehearse before recording. Each piece was taken at least twice, once asking the player to play normally and another with a different expression. Some pieces were asked to be played to exaggerate expressions or cues for synchronization. An annotator with musical training aligned the takes to the score so that the click track associated with the score sounds appropriate for the performance. The data was recorded in two recording sessions, approximately one year apart.

[1] Print Gakufu: `https://www.print-gakufu.com/`

Composer	Title	Genre	Takes
Hakase	Jonetsu Tairiku	Popular	4
Menken	Beauty and the Beast	Popular	4
Bizet	Carmen	Classical	2
Borodin	Polovetsian Dances	Classical	5
Brahms	Hungarian Dance 5	Classical	3
Debussy	Bateau	Classical	2
Elgar	Salut d'Amor	Classical	2
Faure	Berceuse	Classical	2
Mascagni	Cavalleria Rusticana	Classical	3
Mendelssohn	Song Without Words	Classical	4
Pachelbel	Canon	Classical	3
Saint-Saens	Le Cygne	Classical	2
Tchaikovsky	Barcarolle	Classical	5
Tchaikovsky	Troika	Classical	6

Table 1. List of the recorded pieces and the number of takes.

To extract the synchronization parameters, we assumed that one player serves as the human while another serves as the timekeeper. For each distinct note onset event written in the score, a corresponding human onset was obtained by manually aligning to the score and finding the mean onset time of the notes played within a window of the 32nd note of the aligned onset time, repeatedly increasing this radius by 1.1x if no onset exists in the window, and ignoring the note if the window exceeds 0.3 seconds. When the two players do not play simultaneously, we assume that, for the player with no onset, the onset timing is linearly interpolated between the adjacent note onset times of that player. The parameters are estimated using the bGLS method for phase correction models [7], considering the introduction of note values in our formulation. The estimation requires multiple asynchronies, so we compute the parameter by aggregating the current asynchrony and ten neighboring asynchronies. We further smooth the parameters by a moving average window with 4 note onset events to smooth outlier estimates.

4.2 Experiment 1: Parameter estimation accuracy for piece-level cross-validation

We evaluate both the proposed linear (denoted as method **linear**) and DNN models (**DNN**) and two baseline methods that do assume a relationship between the score and SMS parameters. The first baseline estimates a single set of parameters α, β, σ_M, σ_T using the entire *training* data (**uniform-train**), and the second baseline estimates a single set from the entire *validation* data (**uniform-valid**), which can be considered as the theoretical maximal performing method, assuming the independence of SMS parameters on the score.

The models were trained with leave-one-out cross-validation (14-fold cross-validation) at a piece level so that the piece used for training would not be used for evaluation. The models were trained to minimize the mean absolute error (MAE) using Adam. The MAEs of the estimated parameters were then computed for each of the validation folds, and their averages were taken as the metric.

Table 2 shows the result. The proposed method consistently outperforms the baselines. Namely, for α, the MAE

Condition	α	β	σ_T	σ_M
Uniform-train	7.48E-2	2.63E-3	1.89E-2	2.74E-2
Uniform-valid	7.27E-2	2.30E-3	1.81E-2	2.45E-2
Linear	6.90E-2	2.18E-3	1.52E-2	2.46E-2
DNN	**6.68E-2**	**1.70E-3**	**1.38E-2**	**2.33E-2**

Table 2. MAE of the synchronization parameter estimates (piece-level cross-validation).

of the **DNN** method is reduced by 11% compared to uniform, β by 35%, σ_T by 27%, and σ_M by 15%. Incorporating the score is advantageous, as can be seen by comparing **linear**, a simple linear model, and **uniform-valid**, the minimum attainable MAE when ignoring the score. More elaborate models like **DNN** can attain even lower performance. The results are significant for almost all pieces and parameters, using a two-sided t-test with a significance level of 0.01. For condition **linear**, one piece did not show significance for α, one piece for β, one piece for σ_M, and three pieces for σ_T. For condition **DNN**, two pieces did not show significance for σ_M and three pieces for σ_T.

Figure 2 shows an example of the outputs. The ground-truth data contain large variabilities throughout the piece, which cannot be expressed with a constant estimate. Both linear and DNN models capture some of the variabilities, but the linear model tends to revert to the uniform baseline, whereas the DNN model can have lower values.

4.3 Experiment 2: Parameter estimation accuracy for player-level cross-validation

We have repeated the previous experiment, except the cross-validation was performed at the player level. In other words, the piano roll and the SMS parameters obtained from one pianist in the duo were used to train each model to evaluate the MAE of the data from the other pianist and vice versa.

Table 3 shows the result. A similar tendency as the previous experiment can be seen, where the score-dependent parameter estimates have lower MAEs. The differences in the MAE were significant for all pairs of conditions, using a t-test with a significance level of 0.01.

The drop in MAE for **linear** and **DNN** is more pronounced here, suggesting the importance of learning the score over the individual player to estimate the SMS parameters. To elaborate, it is common in a piano duet for the two players to switch musical roles, resulting in the two parts having similar scores dispersed throughout the piece. Thus during training, the model in this experiment was allowed to see a score similar to that in the validation data but not the player in the validation set, whereas the model in the previous experiment was allowed to see the performer in the validation data but not the validation piece. Thus, the fact that MAE is lower for this experiment suggests that it is the learning of the music score (by an arbitrary pianist) that is more important than the learning of the individual player (by an arbitrary piece).

Condition	α	β	σ_T	σ_M
Uniform-train	7.43E-2	2.85E-3	2.05E-2	2.76E-2
Uniform-valid	7.43E-2	2.84E-3	1.71E-2	2.74E-2
Linear	6.56E-2	**1.77E-3**	1.39E-2	2.42E-2
DNN	**5.65E-2**	1.80E-3	**1.19E-2**	**1.82E-2**

Table 3. MAE of the synchronization parameter estimates (player-level cross-validation).

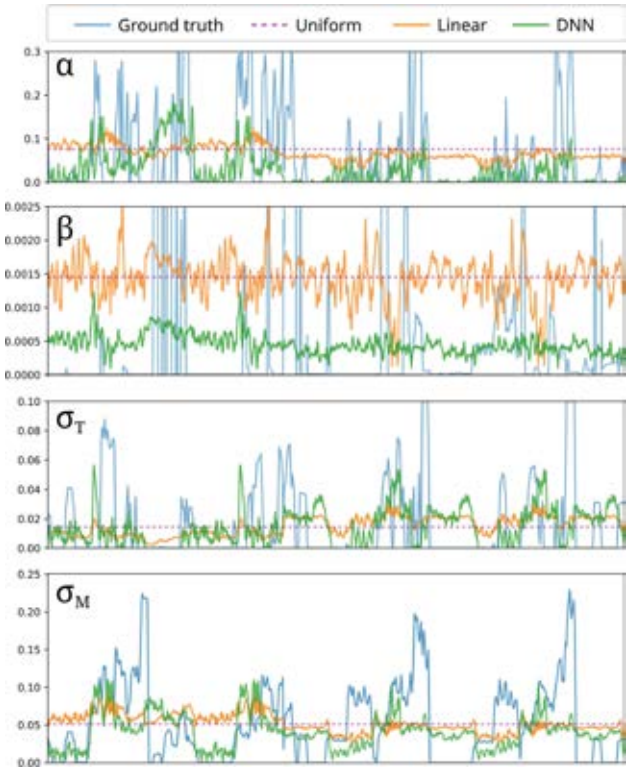


Figure 2. An example of estimated outputs versus the ground truth. Here, "Uniform" refers to condition **uniform-train**, which is almost identical to **uniform-valid** for this piece.

5. DISCUSSION

We have presented a model of SMS that is conditioned on the music score, allowing computers to be mindful of the music score when synchronizing with a human player. A deep neural network model has shown potential in this direction. Due to the simplicity of the parameters, it is possible to assess the model's behavior and combine the insights from the music cognition literature. For example, it has been reported that the behavior of human performers remains consistent at a range of α between 0 and 1 [19], so it is sensible to clip the generated values to ensure an accompaniment system behaves stably for human performers.

The model presented in this method has been utilized as a stochastic state space model for automatic accompaniment systems in a few professional piano duet performances and music installations for the general public. The system, in general, performed stably as long as the score follower functioned properly and the user played more or less within the rehearsed tempo. Strong asynchrony occurs when the user (1) plays with lots of abrupt timing pauses, such as when sight-reading or has motor disabilities, or (2) shows

prominent tempo expression, such as in a ralletando. For the first issue, a better model of non-expert piano performance will become necessary. For the second issue, combining a long-term model of tempo prediction with a local SMS model might be a possible direction.

Our study has a few limitations. First, it cannot express variability in the SMS parameters within a given performance instance. While it is reasonable to assume that the general approach to synchronization between human players remains consistent for a particular music score segment, no two performances are temporally identical, and therefore, no two performances have identical time coordination. This suggests that SMS is also a function of the instantaneous interaction or some auto-regressive stochastic process, which our model currently ignores. Second, since we only had access to one pair of pianists, the across-player cross-validation represents the behavior of players in a single ensemble; it may be possible that different pairs of pianists, or the pianist used in this study paired with another pianist may behave differently. For example, beginners with little exposure to music will mostly likely have different motor noise σ_M, or have different α due to the incapability to attend and listen to others. Our result nonetheless showed that, given a piano duo, it is possible to generalize timing synchronization to unseen pieces by learning from their performances.

6. CONCLUSION

This paper presented an SMS model conditioned on the music score. We demonstrated the effectiveness of incorporating the music score to improve the quality of SMS parameter estimation. Future work includes incorporating music performance features and improving synchronization robustness by piano players of wide skill levels.

7. REFERENCES

[1] A. Cont, J. Echeveste, J.-L. Giavitto, and F. Jacquemard, "Correct Automatic Accompaniment Despite Machine Listening or Human Errors in Antescofo," in *Proc. International Computer Music Conference 2012*, Sep. 2012.

[2] T. Mizumoto, T. Ogata, and H. G. Okuno, "Who is the leader in a multiperson ensemble? Multiperson human-robot ensemble model with leaderness," in *Proc. 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct. 2012, pp. 1413–1419.

[3] A. Maezawa and K. Yamamoto, "MuEns: A Multimodal Human-Machine Music Ensemble for Live Concert Performance," *Proc. 2017 CHI Conference on Human Factors in Computing Systems*, pp. 4290–4301, 2017. [Online]. Available: http://doi.acm.org/10.1145/3025453.3025505

[4] G. Xia, Y. Wang, R. Dannenberg, and G. Gordon, "Spectral Learning for Expressive Interactive," in *Proc. 16th International Society for Music Information Retrieval Conference*, 2015, pp. 816–822.

[5] C. Raphael, "A Bayesian Network for Real-Time Musical Accompaniment," in *Advances in Neural Information Processing Systems*, 2001, pp. 1433–1439.

[6] C. Cancino-Chacón, S. Peter, P. Hu, E. Karystinaios, F. Henkel, F. Foscarin, and G. Widmer, "The accompanion: combining reactivity, robustness, and musical expressivity in an automatic piano accompanist," in *Proc. International Joint Conference on Artificail Intelligence 2023*, 2023.

[7] N. Jacoby, N. Tishby, B. H. Repp, M. Ahissar, and P. E. Keller, "Parameter estimation of linear sensorimotor synchronization models: Phase correction, period correction, and ensemble synchronization," *Timing Time Perception*, vol. 3, no. 1-2, pp. 52 – 87, 2015.

[8] C. Raphael, "Music Plus One: A System for Expressive and Flexible Musical Accompaniment," in *Proc. International Computer Music Association*, 2001, pp. 159–162.

[9] A. Maezawa, "Deep Linear Autoregressive Model for Interpretable Prediction of Expressive Tempo," in *Proc. 16th Sound and Music Computing Conference*, 2019.

[10] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, "VirtuosoNet: A hierarchical RNN-based system for modeling expressive piano performance," in *Proc. International Society for Music Information Retrieval Conference*, 2019.

[11] A. Maezawa, K. Yamamoto, and T. Fujishima, "Rendering music performance with interpretation variations using conditional variational RNN," in *Proc. International Society for Music Information Retrieval Conference*, 2019.

[12] S. Rhyu, S. Kim, and K. Lee, "Sketching the Expression: Flexible Rendering of Expressive Piano Performance with Self-Supervised Learning," in *Proc. International Society for Music Information Retrieval Conference*, 2022.

[13] I. Borovik and V. Viro, "ScorePerformer: Expressive piano performance rendering with fine-grained control," in *Proc. International Society for Music Information Retrieval Conference*, 2023.

[14] B. H. Repp and Y.-H. Su, "Sensorimotor synchronization: A review of recent research (2006–2012)," *Psychonomic Bulletin & Review*, vol. 20, no. 3, pp. 403–452, Jun. 2013. [Online]. Available: https://doi.org/10.3758/s13423-012-0371-2

[15] D. Vorberg and H.-H. Schulze, "Linear phase-correction in synchronization: Predictions, parameter estimation, and simulations," *Journal of Mathematical Psychology*, vol. 46, no. 1, pp. 56–87, 2002.

[16] H.-H. Schulze, A. Cordes, and D. Vorberg, "Keeping synchrony while tempo changes: Accelerando and ritardando," *Music Perception: An Interdisciplinary Journal*, vol. 22, no. 3, pp. 461–477, 2005.

[17] K. Armstrong, J.-X. Huang, T.-C. Hung, J.-H. Huang, and Y.-W. Liu, "Real-Time Piano Accompaniment Using Kuramoto Model for Human-Like Synchronization," in *Proc. 16th International Symposium on Computer Music Multidisciplinary Research*, 2023. [Online]. Available: https://zenodo.org/records/10113439

[18] J. S. Snyder, E. E. Hannon, E. W. Large, and M. H. Christiansen, "Synchronization and Continuation Tapping to Complex Meters," *Music Perception*, vol. 24, no. 2, pp. 135–145, 2006.

[19] B. H. Repp and P. E. Keller, "Sensorimotor synchronization with adaptively timed sequences," *Human Movement Science*, vol. 27, no. 3, pp. 423–456, 2008.

SIMULATING PIANO PERFORMANCE MISTAKES FOR MUSIC LEARNING

Alia MORSI (alia.morsi@upf.edu) [1], **Huan ZHANG** (huan.zhang@qmul.ac.uk) [2], **Akira MAEZAWA** [3], **Simon DIXON** [2], and **Xavier SERRA** [1]

[1] *Music Technology Group*, **Universitat Pompeu Fabra**, Barcelona, Spain
[2] *Centre for Digital Music*, **Queen Mary University of London**, London, United Kingdom
[3] **Yamaha Corporation**, Hamamatsu, Japan

ABSTRACT

The development of machine-learning based technologies to support music instrument learning needs large-scale datasets that capture the different stages of learning in a manner that is both realistic and computation-friendly. We are interested in modeling the mistakes of beginner-intermediate piano performances in practice or work-in-progress settings. In the absence of large-scale data representing our target case, our approach is to start by understanding such mistakes from real data and then provide a methodology for their simulation, thus creating synthetic data to support the training of performance assessment models. The main goals of this paper are: a) to propose a taxonomy of performance mistakes, specifically apt for simulating or reproducing/recreating them on mistake-free MIDI performances, and b) to provide a pipeline for creating synthetic datasets based on the former. We incorporate prior research in related contexts to facilitate the understanding of common mistake behaviours. Then, we design a hierarchical mistake taxonomy to categorize two real-world datasets capturing relevant piano performance contexts. Finally, we discuss our approach with 3 music teachers through a listening test and subsequent discussions.

1. INTRODUCTION

To build music education systems that can detect different types of performance mistakes we must devise frameworks that represent such mistake patterns effectively. Analyzing beginner-intermediate piano performance data from practice settings to understand and model learner behaviours would be useful, but to our knowledge there are no published datasets with annotations specifically capturing this context. Furthermore, it remains an open question what type of annotation would be both realistic and computationally friendly, and what its relationship would be to the learning behaviours we should seek to detect. Our goal is to develop an understanding of such mistakes and provide a methodology for their realistic simulation, to provide relevant synthetic data for model (pre-)training.

In computational research on piano performance mistakes, there is a tendency to discuss mistakes in terms of literal deviations from the underlying music score, such as pitch insertions and deletions [1–3], and rhythmic deviations [2], with the latter being less frequently examined because pitch is relatively fixed by the compositional notation of Western tonal music [4] while rhythmic errors are more difficult to observe under score-performance alignment algorithms [5]. While this paradigm permits interesting analysis of performance mistakes, especially when coupled with musical and psychological perspectives, it alone is not ideal for recreating them in a realistic manner.

We propose a hierarchical, computation-friendly taxonomy of piano performance errors based on observations from two piano performance datasets related to our target case, where mistake behaviours are higher level units formed from literal score deviations on the axes of pitch, time, velocity, and structure (which we consider lower-level). Then, we create a performance mistake simulation model based on this taxonomy and use it to apply mistakes to MIDI piano performances belonging to a suitable repertoire with respect to the targeted proficiency levels. Finally, we present results from a questionnaire and interview study with music teachers, who provided feedback on the realism of our synthesized examples, taxonomy, and methodology.

The paper is organized as follows: section 2 builds a foundation to understand piano performance mistakes by connecting with related research regarding their causes, manifestations, and categorizations. In section 3 we demonstrate our main observations from real performance data, after which we define our hierarchical framework for mistake analysis and simulation in section 4. Sections 5 and 6 concern the implementation of our mistake simulator, and the preparation of MIDI performances for use as input. The results and insights of the teacher interviews are shown in section 7, followed by our conclusion and future work (section 8). All code and materials are available on the companion page [1].

2. PIANO PERFORMANCE MISTAKES

Technical difficulties, lack of concentration, and poor memorization are some of the many factors which contribute to performance errors [1, 6]. Some research further examines patterns of performance mistake production with respect to their underlying psychological processes [4, 7, 8], where

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] https://github.com/Alia-morsi/piano-synmist

insights on a performer's mental representations of a music score can be provided through analyzing the distribution and relative frequencies of performance mistakes [2].

Since the main purpose of this work is to simulate performance mistakes based on a realistic taxonomy, this section reviews the appearance of performance mistakes (and hence how they are detected, and potentially reproduced), some of their prominent patterns, and how different researchers have categorized them.

2.1 Appearance of Performance Mistakes

In most research on piano performance mistakes, determining their presence or their extent is based the faithfulness of a performance to its corresponding music score [1]. One limitation of this approach is that it treats all deviations from the music notation as performance errors, despite such deviations (ornamentation, rubato, etc) being expected as part of a performer's artistic license in Western tonal music [4]. Furthermore, it does not account for the fact that not all score deviations are perceived as errors by listeners [1], nor for the differences between editions of scores.

2.1.1 Pitch Insertions, Deletions, and Substitutions

In research relying on such deviations for analysis, whether in music technology contexts [3,5,6,9,10] or psychological studies [1,4], performance mistakes are often reported in terms of the edit distance between score and performed note sequences, i.e. as pitch insertions, deletions, and substitutions, although often the latter is just treated as a simultaneous pitch deletion and insertion. In some cases, percentages of these note-edit categories are reported [1, 2, 6]. Such statistics can be useful to observe quantitative relationships between performances of music pieces with similar characteristics (e.g. style, technical difficulty). Flossman et al. [6] investigated whether the ratios found by Repp [1] are preserved across other data (ballade, polonaise, nocturne, etc.). Though without context those statistics might not be very informative on their own which is why researchers often propose further categories as shown in 2.2. Moreover, for the purpose of simulating mistakes, it is unlikely that applying these same percentages even on the same pieces would result ones that sound natural.

Despite the importance of timing and velocity in the perception of mistakes, they have not received as much attention as pitches. We believe that this could be due to the relative ease of pitch quantization and categorization compared to timing, as there could exist confusions between expressive timing and mistake behaviors (fermata vs hesitation, and rubato vs tempo instability, etc). Another reason could be due to variability in note velocity measures. Both should receive more attention in subsequent performance mistake studies with recent advances in Automatic Music Transcription for piano with respect to temporal resolution [11] and velocity estimation [12].

2.1.2 Reproducing Piano Performance Mistakes

To the best of our knowledge, apart from the work of Morsi et al. [13] where the simulation of performance mistakes was conducted as data augmentation, there has not been work specifically addressing the reproduction or simulation of performance mistakes. Despite differences in naming, the score deviations applied in their system were: 1) pitch omission, 2) pitch substitution, 3) pitch insertion, 4) short pause (between 0.3 and 0.8 seconds) and repeat last note, and 5) long pause (between 2 to 4 seconds) and repeat last note. Insertions and substitutions were constrained to be n semitones (specified by the user) around the correct note.

2.1.3 Common Patterns and Behaviours

Several researchers affirm the impact of musical context on the type of errors [1, 2, 4, 7]. For example, harmonically related errors more associated with homophonic than polyphonic passages [4], and pitch errors mostly occur in non-melody voices or inside chords [1, 2]. Furthermore, less errors are observed in notes of recurring musical motifs than for other passages [2], and higher error frequencies are observed with higher note densities in a piece, reflecting the increased technical demands the more the notes that should be played per time unit.

Other patterns could be related to the context of a performance (i.e sight reading vs 'quick study' vs recital) and performer characteristics such as age [8] or performance level [14]. For example beginners are more likely to repeat errors, either because they do not realize there was an error, or because they are unable to correct it in repeat performances [14]. Children with more musical training are better able to detect and correct performance mistakes with less perseveration behavior [15]. Morijiri et al. [16] note that beat interruptions were the most common type of errors among adult beginners. Weber and Parncutt [14] propose a theory for error management in music performance, where a whole error undergoes *pre*, *during*, and *post error* stages. A performer's *pre-error* behavior is a form of risk management, while the *during* and *post-error* stages are forms of error management. This connects with the observation that the majority of insertion errors are of low note velocity compared to their immediate neighbourhood [17], as a performer's awareness that they are about to make a mistake causes them to play the inserted note at a lower velocity as a form of risk management. It also suggests that it might be useful to consider the connection between consecutive mistake events so that the behaviours of the three stages can be effectively demonstrated.

2.2 Categorizations of Performance Mistakes

Researchers have proposed different categories to contextualize the deviations between the notes of a performance and its score (described in section 2.1.1) into more descriptive categories based on the performance context, to better identify performance mistakes and compare between error-trends. Palmer and van de Sande [4] have categorized errors on the dimensions of size (note, chord, or a combination), source (contextual/non-contextual), type (insertion, deletion, substitution, shift), and movement (anticipatory/perseveratory). In another example on children piano sightreading, Gudmundsdottir [8] classified the pitch insertions observed as erroneous pitches (those which do not match target pitches in the score) and redundant pitches

(repetitions of correct pitches, which could be due to hesitation).

In further analysis of the Magaloff Corpus [6, 18], the following error categories were proposed, which are mostly pitch related by virtue of their context for mistake analysis given that this corpus represents on-stage behavior by a professional pianist: **1) Forward-Related Error**: an 'out-of-score' choice due to the influence of an upcoming note; **2) Backward-Related Error**: the same, but due to the influence of a previous note; **3) Repeated Note**: an example of a backward-related error where a note is played twice, with the second note often played weaker than the first; **4) Nonharmonic Error**: a note insertion that introduces a harmonic clash; **5) Harmonic Error**: an insertion that does not introduce a harmonic clash, which usually signals a memorization problem, although occasionally it could be deliberate, for harmonic emphasis; **6) Tied Note**: Could take the form of a technical simplification applied when a repeated note is held like a tied one (a deletion), or the form of a repeated note instead of a tied one whether due to a memorization problem or intentionally; **7) Systematic Error**: an error that occurs in the same context more than 60% of the time it is observed; **8) Note Order Error**: occurs when switching the playing order of two or more successive notes; **9) Omitted Inner Voice**: deletions that involve inner voices. It is worth noting that errors can belong to more than one category.

3. EMPIRICAL OBSERVATIONS FROM PERFORMANCE DATA

We use two datasets to investigate performance errors. The *Burgmüller* set [13] consists of 50 recordings (25 pieces recorded twice) from the *Burgmüller Etudes, Op.100*, performed by an advanced pianist who only studied the Etudes briefly before the recording. The context of the recordings allows for natural mistake behaviours such as repetitions and occasional pauses, unlike well-practiced performance settings. The dataset presented by Jiang [19], which we call the *Expert-Novice* set, includes 83 piano performances from 21 adult beginner players, with a repertoire covering 7 easy pieces of folk and pop songs in the Western tonal tradition, with some rhythmic syncopation. It includes beginner-level mistakes in a recital context.

We analyzed both the *Burgmüller* and *Expert-Novice* datasets to better describe, represent, and simulate performance mistakes. Both sets contain natural mistakes since none of the performers were instructed to make mistakes deliberately. The result of this process is the basis for the proposed framework in section 4.

3.1 Burgmüller Dataset

Although the mistakes in *Burgmüller* are not beginner-intermediate level due to the experience of the performer (as noted in the feedback from music teachers in section 7), we believe that practice-like behaviours can be observed given its collection context (non-recital setting with limited time given to the performer to learn the pieces).

The performances are provided in MIDI format with annotations of the sites of mistakes. We manually listen to the mistakes and inspect their corresponding music scores. We analyze the first recordings of Etudes 9, 10, and 17-20 (each of the 25 Etudes was played twice), and provide our raw observations on the paper's companion page.[1]

The most commonly observed mistake is what we refer to as a **note mistouch**, which is an extra note inserted concurrently with a score note a tone or semitone above or below it. According to the categories explained in section 2.2, this would most often be of a *nonharmonic* nature, but also could arise as a *forward-related* or *backward-related* error, or a even a *systematic* one if it occurs multiple times over the course of the performance. A less frequent error type is a **note substitution**, where an incorrect note replaces a correct one. The substitution could a *harmonic error*, or a *nonharmonic* one (often due to confusion of accidentals). In some cases the substituted note is played strongly, indicating confidence of the player, and in other cases the substitutions demonstrate the *pre-error* behaviour described in section 2.1.3.

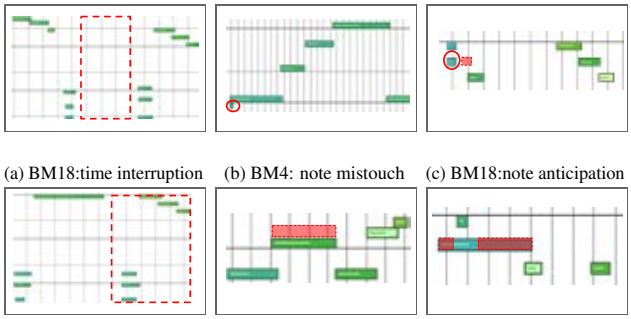
From a temporal perspective, we observe **time interruptions** which can manifest as long or short pauses in unlikely locations that interrupt the musical flow, which can sometimes occur after another mistake. Another pattern is that of holding a note or chord longer than expected, and causing a delay in the onsets of the upcoming score notes. We refer to this pattern as a **dragging** note. Sometimes it is observed on a correct score note, hence manifesting as irregularity in note times and velocities, and sometimes the dragging note behaviour happens with an incorrect note.

Note repetitions are also quite common. Often they are either a repeat of the last played note, or a repeat from a reasonable prior point in the score (e.g. the previous beat). Such repetitions often occur as a reaction to making a mistake, whether it is a time interruption, note substitution, a mistouch, or another. We also observe **joined notes**, where a player would tie a note instead of playing it twice (referred to as *tied note* in section 2.2, and occasional **note deletions**. Some examples are shown in Figure 1.

3.2 Expert-Novice Dataset

Since the performances are provided as audio, we transcribe [11] and manually correct the performances (pipeline available [1]) to obtain MIDI versions for easier inspection. We observe similar mistake categories as those found in *Burgmüller*.

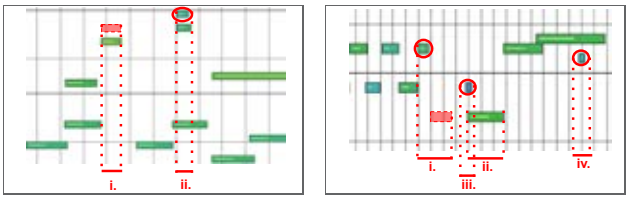
Note mistouches are observed around running scales or rapid chord changes. **Note substitutions** mostly occur with melody notes, often around accidentals, but also less frequently within the context of chords or harmonic intervals (2 note chords). The substitutions could be affected by the previous or next harmony, making them *backward-related* or *forward-related* errors respectively. An additional type of extra note was specifically observed in *Expert-Novice* with the addition of a note horizontally on an extra time step. It it is not clear if the mistake is a note substitution and a repeat, or due to mis-memorization and incorrect sense of the rhythmic grid. Both could be likely given the performances are beginner-level. Finally, **note deletions** take place in a less subtle manner than in *Burgmüller*, with



(a) BM18:time interruption (b) BM4: note mistouch (c) BM18:note anticipation

(d) BM18: note repetition (e) BM20:note substitution (f) BM14: joined notes

Figure 1: Illustrations of some examples from analysis of the Burgmüller data. Each subfigure is captioned with Etude number and error type. Red highlights the mistake.



(a) BM20: (i) a note substitution followed by (ii) a mistouch.

(b) BM16: (i) a note substitution followed by (ii) a mistouch, (iii) a time interruption and (iv) a note mistouch.

Figure 2: Note substitutions leading to subsequent mistakes.

either a melody note or an entire chord missing from the performance.

Time interruptions (or silences) and **dragging** notes are also observed, where in some cases the interruptions are followed by **note repetitions**. **Rhythmic errors** are different from dragging, occurring when one musical event gets displaced, or is performed with a wrong duration, but still fits into the rhythmic grid. This type of error usually occurs consistently within the piece, likely originating from practice. The more jazzy and syncopated repertoire shows more timing mistakes.

4. HIERARCHICAL TAXONOMY FOR MISTAKE ANALYSIS AND SIMULATION

From the analysis in section 3 we conclude that it is infeasible to isolate the detected mistake behaviours solely as either pitch, time, velocity or structure related mistakes, as most of the encountered error categories would manifest in terms of the same granular score deviations with slightly different parameters. For example, *time interruptions* (Fig. 1a) can be implemented as the insertion of a time gap through a shift t in the time of the performance MIDI roll, and a repetition error would also be formed of the same time shift operation with additional note insertions (Fig. 1d). The parameters of the inserted notes could be determined based on the behavioural profile of the player, how they are expected to repeat, and the effect of making a mistake on further mistakes until the player recovers, as shown in Fig. 2.

Taking into account the previous conclusions, we propose a multi-level taxonomy for understanding and simulating performance mistakes. It separates between **low-level** operations (granular deviations between a performance and its music score), and conceptual mistake categories such as those highlighted in section 2.2 which we refer to as **mid-level**. Hence, mistakes are mid-level concepts comprised of one or more lower level operations. Mid-level mistake categories should be chainable to form meaningful behaviours, to allow the construction of higher level concepts would permit the rendering of realistic student performances based on descriptions of student profiles or emotional states.

This taxonomy would be relevant for both the automatic analysis of performance files with mistakes or for the simulation of errors on mistake-free MIDI performances, though in this paper we only do the latter. For analysis, the goal would be to use computational approaches like audio-to-score alignment to detect the low-level deviations and connect them to the proposed mid-level categories. For mistake simulation, implementations of the desired mid-level mistakes should be provided using low-level operations.

Although we do not surpass the mid-level, we envision that higher levels of this taxonomy would allow the simulation of performances that reflect the different traits of learners using combinations of the proposed mid-level mistake concepts. For example, students who demonstrate impatience through pursuing tempi that they cannot really manage would run into frequent mistouch, substitution, and rollback errors, or less confident students would be likely to produce more ghost notes and joint notes caused by their nervous touch of the keys. Simulations at this level are analogous to training a model that generates text in the style of a specific writer, or to compose music in the style of a particular composer. As discussed in section 7, the piano teachers have encountered examples where student attitudes affect the patterns of mistakes they commit. This direction would require obtaining and comparing performance data of different students, which we hope to conduct in the future.

4.1 Low Level: Granular Score Deviations

They are operations directly applicable on a MIDI file with each affecting a single dimension or axis operations. **a) pitch insertion** and **b) pitch deletion** are the most basic, which are adding or removing pitches with respect to the reference. We treat substitutions as higher level operations that constitute a deletion and an insertion. **c) delay note** and **d) anticipate note** are adjustments to the note onset time. **e) extend note** and **f) shorten note** adjust the duration via the note offset time. Further temporal operations are **g) shift time**: insert a gap in the score; **h) go back**: move the 'playhead' back to a past point in the reference; and **i) skip**: skip a portion of the score.

4.2 Mid Level: Mistake Categories and Behaviours

The categories proposed at this level are what we would use to describe the mistake behaviours in section 3. Each is composed of one or more of the low level deviations described in section 4.1, and is connected to musical context/texture. We define 2 classes of mistake behaviours: **recovery operations** and **core mistakes**. Recovery operations encompass how players would regain the flow in

their performance, whether it is due to a loss of concentration, a memorization error, or a response to another kind of error until recovery. Core mistakes are the main mistake events, such as accidental note mistouch, or incorrect note asynchrony.

4.2.1 Recovery Operations

Rollback is when a player repeats a section that has already been played. It consists of a *shift time* and several *pitch insertion* operations. Sometimes rollback affects only the hand with a mistake, or sometimes both hands. The velocity and time of the repeated notes can vary depending on the target behaviour. Rollback usually follows a *re-orientation*, which in some cases follows a *core mistake*.

Re-orientation is the behaviour expressed when a player is in the process of regaining their playing-flow, usually in response to committing an error, but could also be due to a need for concentration due to the demands of the piece performed. It consists of a *shift-time* (reflecting either a short or long pause), and potentially one or more *extend-notes* until the flow is resumed, and could be followed by a *rollback* before the flow is resumed.

4.2.2 Core Mistakes

Mistouches are extra notes inserted concurrently with a score note a tone or semitone above or below it. On their own, they do not interrupt flow (due to their vertical co-occurence with a correct note). However, they can affect subsequent notes (resulting in error chaining), or could be followed by one or more recovery operations. From our observations, they commonly occur after jumps (or leaps), especially with staccato, or during runs (such as octave runs).

Ghost Notes are notes that are hit too lightly, which results in a short note with an almost inaudible volume. In the teacher feedback (section 7), they identified this error as common with students who tend to lack confidence, and mostly happens with notes played with the 5th (little) finger in an arpeggio or a chord. One also noted that this error is associated with a high reorientation time with beginner students (as they are trying to find the source of the sound discrepancy), and is likely followed by a rollback.

Incorrect Asynchrony is when a set of notes played together have onsets starting at slightly different times. This translates into several small **delay note** or **anticipate note** operations. Asynchrony is often used as an expressive tool, and we do not have a formal definition of when it is considered an error or not.

Substitution Error is when a player hits the wrong note/s with confidence, whether due to a mis-memorization or a wrong hand movement. Such substitution could be *harmonic* (e.g. a musically appropriate interval above or below the score note) or *nonharmonic* (the substitute note does not fit the musical context, like a misread accidental). Such an error could affect one note, two notes, or the whole chord.

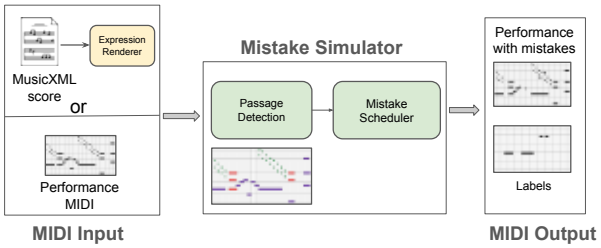


Figure 3: Mistake simulation pipeline. In the passage detection illustration, green and red stand for double notes and block chords, respectively.

Anticipated Note is when a correct note is played earlier than expected. This could be due to mismemorization of the rhythm, or simply due to accidentally touching the next note when playing a sequence. Again, this could also be an expressive tool, so not every anticipation should be considered wrong. In some cases, anticipated notes arise from a mistouch.

Join Notes occur when two consecutive strokes of the same key become one longer press. It fulfills both the *forward-related* error and the *tied notes* categories from [6], and can be executed as a combination of **extend note** and **delete note** lower level operations.

5. MISTAKE SIMULATOR IMPLEMENTATION

We develop a pipeline (Fig. 3) for applying the mid-level mistakes on mistake-free MIDI performances based on 2 steps: **1) Musical Element Detection**, which analyzes the score to identify relevant musical elements or passages for the application of mistakes, and **2) Mistake Scheduling**, which applies mid-level mistake operations on the MIDI performance (see section 5.2.1). The mistake scheduling phase is responsible for determining which mid-level mistakes to apply and their parameter values based on the output of step 1. The supported mistake operations are explained in section 5.2.1. They are a subset of the mid-level behaviours (section 4.2) and are built on an implementation of the low-level score operations described in section 4.1. Our code relies on the Partitura package [20]. As input, the Mistake Simulator expects a mistake-free MIDI performance, either a rendered performance from a music score or a real performance. We apply VirtuosoNet [21] to render a curated set of music scores within the skill level of our target user group (see section 6). The real performances we use are from the Vienna 4x22 [22] and SMD [23] datasets. The output of this pipeline are 2 MIDI tracks for each piece: one with the modified performance MIDI with mistakes, and another with labels corresponding to the applied mistake operations (both low and mid level), as shown in Fig 3.

5.1 Musical Element Detection

We implemented a basic rule-based system to detect regions of notes in a MIDI performance as **harmonic intervals** (2

note chords), **scales** and **block chords**. We chose these elements as simple examples of locations where it could be interesting to apply mistakes, although it is not by any means a thorough list.

Harmonic Intervals: For each note n, we obtain its set of onset neighbours with a diameter of d semitones $N_{n,d,\theta} = \{|Onset(k) - Onset(n)| < \theta, |Pitch(k) - Pitch(n)| <= d, k \in K\}$ where θ is the timing threshold which is set as 50ms. If $\|N_{n,d,\theta}\| = 2$ (only two notes within this onset window), then we label n as part of a harmonic interval.

Block Chords: Similarly, for the block chords we search for the neighbours with approximately the same onset and offset window where $\|N_{n,d,\theta}\| > 2$.

Scales: We look for the consecutive neighbours of n: $N_{n,d,\theta} = \{Onset(k) - Offset(n) < \theta, |Pitch(k) - Pitch(n)| <= d, k \in K\}$. We used a diameter d value of 2, although the value should be adjusted if this is to account for scales that include wider intervals.

5.2 Mistake Scheduling

In addition to the result of the mistake element detection step, the mistake scheduler expects a probability table (e.g. $P_{\text{Mistouch}}(n|n \in \text{BlockChords}) = 0.4$) which assigns a heuristic distribution for each mistake per musical element. Given a target number of mistakes per element class, the scheduler samples the notes on which mistakes will be applied.

5.2.1 Mistake Operations

Of the mid-level operations of section 4, we implement one recovery operation and four core mistake operations. **Rollback**(n, t): Starting from the onset of n for a subsequent (user-defined) window of t seconds, all note events are repeated, while all subsequent notes outside the window are shifted by t. **Forward-backward insertion**(m, n): A forward or backward neighbour m of n will be inserted on top of note n with similar onset and offset. **Mistouch**(m, n): A mistouched note m will be inserted on top of n with similar onset and offset times, where $|Pitch(m) - Pitch(n)| < 2$. **Note Substitution**$(m, n)$: n will be replaced by a pitch alteration m. In the case of chords, the whole chord group is replaced with the new pitches. **Dragging Note**(n, t): Starting from the onset of n, all note events are shifted by t seconds. At the end of the pipeline two parallel MIDI tracks can be exported: a) the modified performance and b) labels reflecting all the operations applied, in terms of low level operations and mid-level mistakes.

6. PERFORMANCE MIDI PREPARATION

6.1 Rendered Music Scores

We filter a set of piano-learning repertoire and download corresponding score renditions from Musescore [2] . To obtain expressive performances of the MIDI files, we applied the open-source performance rendering model VirtuosoNet [21] to the scores. Overall, we gathered 153 music scores, with a rendered duration of 208.16 minutes. Further details are provided on the companion page.[1] Our scores included the following: 1) Notable piano education works *Alfred's Basic Adult All-in-One Course (Book 1-2)* [24] and *Piano Adventures by Nancy Faber (Book 1-2)* [25]. 2) *Burgmüller Op.100* for comparison with the Burgmüller performance set (section 3.1 **3)** Samples from three Czerny etudes (*Op.599, Op.849, Op.299*). **4)** A compilation from catalogues of easy-to-intermediate level classical masterpieces, compiled by Schirmer Library [3] and ABRSM examination board [4] , to form the *Easy Classical* collection. Note that some of the pieces are custom reductions or rearrangements of famous pieces. All public-sourced MusicXML scores have been manually inspected. For the custom reduction / rearrangement scores from the easy classical selections, we validate them by playing to ensure they are within a beginner-intermediate level range.

6.2 MIDI Performance Data

We use subsets of the ASAP [26] and SMD [23] datasets filtered by heuristics to eliminate performances that would certainly be out of reach for the beginner-intermediate range: Average note density (notes per second) ≤ 10; Performance length ≤ 150 seconds; Number of polyphonic voices ≤ 3 (this is verified manually, by removing all the fugues from ASAP, and checking the orchestration for SMD).

Since ASAP contains multiple interpretations of the same piece, we only count one for each composition to avoid redundancy. Overall, this brings 32 pieces from ASAP and 11 pieces from SMD to constitute our score collection. We also use the Vienna 4×22 corpus [22] that features four excerpts with slower passages from the classical repertoire.

7. FEEDBACK FROM MUSIC TEACHERS

We conducted a 90 minute interview with three music teachers, each consisting of a listening survey to rate the realism mistake excerpts and an open-ended discussion about common student mistake patterns and our proposed framework.

7.1 Part 1: Mistake Excerpt Ratings

The teachers rated 12 short mistake excerpts on a scale of 1–5, where 1 denotes a mistake that sounds artificial, and 5 denotes one that sounds like a real student mistake. They were asked to justify each choice, thus providing insights into the potential shortcomings of our approach. The samples included two real mistake excerpts from the Burgmüller dataset, two synthetic mistakes applied on real performances (sec. 6.2), and eight synthetic mistakes on rendered performances (sec. 6.1). The full study as well as the teacher responses can be found in the project repository.[1]

The rating results demonstrated a degree of agreement between the teachers (standard deviation for all samples averaged to 0.65), suggesting a common perception of student mistake patterns. The highest average rating for an excerpt was 4.33, with the perceived naturalness owing to a confusion before the mistake which they thought reflected real student behaviour. The worst excerpt received a 1.33 average rating, and the reason given by one of the teachers was that the high pitched notes are difficult to get wrong when

[2] musescore.org

[3] https://www.halleonard.com/series/SCHLIB/schirmers-library-of-musical-classics

[4] https://www.abrsm.org/en-es

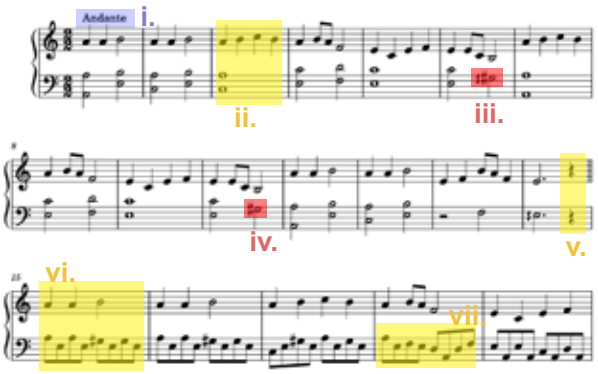


Figure 4: Teacher labeling of common mistakes on *Sakura*, from Alfred's [24] book 2: i) Tempo misread; ii) LH rhythm might erroneously double right hand; iii) and iv) Accidental misread; v) Rest duration not being properly executed; vi) Change of LH duration leads to wrong rhythm; vii) Complex structure, thumb crossing may lead to delays.

in playing in the mid registers. The teachers demonstrated most discrepancy in an excerpt from *Clair de Lune*, which was rated 5 by a teacher and 1 by the others. The low raters thought the mistakes were too drastic such that a natural behaviour would have been to rollback rather than continue. On the other hand, the teacher that rated it highly found the confident recovery sensible given that it happened with the motif, something the player should have internalized well.

Interestingly, they commented that real Burgmüller excerpts did not always sound like natural student mistakes (with the lower rated example receiving an average rating of 2.67), mainly due to the confidence demonstrated by the performer even when committing and recovering from mistake. To them, this behaviour rather sounded as if an advanced performer was deliberately inducing mistakes in their performance.

During the test, the teachers expressed a difficulty of decoupling between evaluating the naturalness of a mistake from the unnaturalness of the expression and phrasing compared to their expectation of a student performance, worrying that a suspicion that an excerpt is synthetically rendered might affect their judgement. However, the two examples of human-recorded performances with synthetic mistakes do not have consistently higher ratings than the rendered ones,

7.2 Part 2: Open Discussion on Mistake Taxonomy and Common Student Error Patterns

In general, the teachers found the proposed mistake categories sensible. As highlighted by their responses to the questionnaire, the recovery from a mistake significantly affected whether an excerpt was perceived as natural or artificial, which validated our dedication of a separate *Recovery* category for the mid-level mistakes. However, they noted that it was challenging for them to think in terms the exact deviations constituting a mistake, because their teaching typically focuses on demonstrating how to improve an incorrectly performed region through highlighting correct techniques, articulations, and reinforcing higher-level musical concepts like phrasing and dynamic contrasts.

Additionally, they emphasized the importance of score agreement in early learning stages where students might be unaware of their reading errors. We asked them to mark mistake-prone regions on beginner scores, and they highlighted areas that included altered accidentals, changes of tempo markings or note durations and rests, as shown in Fig 7.2. They also mentioned that the playing context influences mistake expectations, with a higher presence of rollback behavior during practice sessions due to repeated attempts to correct fragments, as opposed to other situations where the focus is on continuity, like run-throughs or recitals.

Finally, they provided examples where student attitudes or personalities affected their mistakes and performance habits. Some students might demonstrate an insistent character (hammer the keys really hard as they try to to fix the mistakes, reinforce the correct playing with high-energy, show few hesitations); or be 'unsure' (very insecure about the note, produce ghost note that hits the key very slightly); or 'rushing' (struggles to keep the tempo or rhythmic flow, frequently resulting in mistakes up when playing with two hands). Furthermore, they noted the effect of player age (adults vs children) on rollback behaviour, and the difference between the mistakes made by advanced players on familiar pieces and by students on newly-learned pieces. These observations connect with those in Section 2.1.3, both suggesting that for our pipeline to create coherent full simulated performances it would require connecting between the target behaviour, its effect on the mistake parameter values, and the connections between consecutive mistakes.

8. CONCLUSION AND FUTURE WORK

In this paper, we introduced a hierarchical taxonomy for categorizing performance mistakes based on the analysis of the *Burgmüller* and *Expert-Novice* datasets, and developed a pipeline for simulating these mistakes on error-free piano MIDI files. We validated our taxonomy with music teachers and gained their support. Looking ahead, we aim to analyze more performances to better understand the correlations between different types of mistakes and their parameters, and explore how various musical elements influence these errors. This research marks an initial step towards creating a comprehensive dataset for developing machine-learning applications in music education.

Acknowledgments

We would like to thank our participating music teachers: Alexander von Oppenbach, Davide Ugalde, and Pablo Puentes for their invaluable feedback and enthusiasm. This research was partially supported by the project Musical AI - PID2019-111403GB-I00/AEI/10.13039/501100011033, funded by the Spanish Ministerio de Ciencia e Innovación and the Agencia Estatal de Investigación, and the UKRI Centre for Doctoral Training in Artificial Intelligence and Music (grant number EP/S022694/1).

9. REFERENCES

[1] B. H. Repp, "The Art of Inaccuracy: Why Pianists' Errors Are Difficult to Hear," *Music Perception: An*

Interdisciplinary Journal, vol. 14, no. 2, pp. 161–183, 1996.

[2] B. Gingras, C. Palmer, P. N. Schubert, and S. McAdams, "Influence of Melodic Emphasis, Texture, Salience, and Performer Individuality on Performance Errors," *Psychology of Music*, vol. 44, p. 847–863, 2016.

[3] E. Benetos, A. Klapuri, and S. Dixon, "Score-informed Transcription for Automatic Piano Tutoring," in *European Signal Processing Conference*, 2012.

[4] C. Palmer and C. van de Sande, "Units of Knowledge in Music Performance," *Journal of Experimental Psychology: Learning, Memory, and Cognition*, vol. 19, no. 2, pp. 457–470, 1993.

[5] E. Nakamura, K. Yoshii, and H. Katayose, "Performance Error Detection and Post-Processing for Fast and Accurate Symbolic Music Alignment," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[6] S. Flossmann, W. Goebl, and G. Widmer, "The Magaloff Corpus: An Empirical Error Study," in *Proceedings of the 11th International Conference on Music Perception and Cognition (ICMPC)*, 2010.

[7] C. Palmer and C. van de Sande, "Range of Planning in Music Performance," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 21, p. 947–962, 1995.

[8] H. Gudmundsdottir, "Pitch Error Analysis of Young Piano Students' Music Reading Performances," *International Journal of Music Education*, vol. 28, pp. 61–70, 2010.

[9] Y. Jiang, F. Ryan, D. Cartledge, and C. Raphael, "Offline Score Alignment for Realistic Music Practice," in *Proceedings of the 16th Sound and Music Computing Conference (SMC)*, 2019, pp. 387–393.

[10] T. Fukuda, Y. Ikemiya, K. Itoyama, and K. Yoshii, "A Score-Informed Piano Tutoring System with Mistake Detection and Score Simplification," in *Proceedings of the 12th International Conference in Sound and Music Computing (SMC)*, 2015.

[11] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-Resolution Piano Transcription with Pedals by Regressing Onset and Offset Times," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[12] H. Kim and X. Serra, "DiffVel : Note-Level MIDI Velocity Estimation for Piano Performance by A Double Conditioned Diffusion Model," in *16th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Tokyo, Japan, 2023.

[13] A. Morsi, K. Tatsumi, A. Maezawa, T. Fujishima, and X. Serra, "Sounds Out of Pläce? Score-Independent Detection of Conspicuous Mistakes in Piano Performances," in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, 2023.

[14] S. Kruse-Weber and R. Parncutt, "Error Management for Musicians: an Interdisciplinary Conceptual Framework," *Frontiers in Psychology*, vol. 5, 2014.

[15] C. Palmer, "Music Performance," Tech. Rep., 1997. [Online]. Available: www.annualreviews.org

[16] Y. Morijiri, O. Satoshi, A. Maezawa, and T. Fujishima, "Understanding the challenges for adult beginners at piano practice from an analysis of errors," *Proceedings of the 13th Asia-Pacific Symposium for Music Education Research, Virtual Conference*, vol. 28, 2021.

[17] S. Flossmann, W. Goebl, M. Grachten, B. Niedermayer, and G. Widmer, "The Magaloff Project: An Interim Report," *Journal of New Music Research*, vol. 39, no. 4, pp. 363–377, 2010.

[18] S. Flossmann and G. Widmer, "Toward a Model of Performance Errors: A Qualitative Review of Magaloff's Chopin," in *International Symposium on Performance Science*, 2011.

[19] Y. Jiang, "Expert and Novice Evaluations of Piano Performances : Criteria for Computer-Aided Feedback," in *Proceeding of the 24th International Society on Music Information Retrieval (ISMIR)*, 2023.

[20] C. Cancino-Chacón, S. D. Peter, E. Karystinaios, F. Foscarin, M. Grachten, and G. Widmer, "Partitura: A Python Package for Symbolic Music Processing," in *Proceedings of the Music Encoding Conference (MEC)*, Halifax, Canada, 2022.

[21] D. Jeong, T. Kwon, Y. Kim, K. Lee, and J. Nam, "VirtuosoNet: A Hierarchical RNN-based System for Modeling Expressive Piano Performance," in *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, Delft, Netherlands, 2019.

[22] W. Goebl. (1999) The Vienna 4x22 Piano Corpus. http://dx.doi.org/10.21939/4X22.

[23] V. Konz, W. Bogler, and V. Arifi-M, "Saarland Music Data," *Late-Breaking and Demo Session of the International Society on Music Information Retrieval (ISMIR)*, 2011.

[24] M. Manus, A. V. Lethco, and W. A. Palmer, *Alfred's Basic Adult All-in-One Course*. Alfred Music, 1994.

[25] N. Faber and R. Faber, *Faber Piano Adventures*. Faber Piano Adventures, 1996.

[26] F. Foscarin, A. Mcleod, P. Rigaux, F. Jacquemard, and M. Sakai, "ASAP : A Dataset of Aligned Scores and Performances for Piano Transcription," in *Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR)*, 2020.

TABLATURE GENERATION FROM LEAD SHEETS FOR FINGER-STYLE SOLO GUITAR

Shunsuke SAKAI[1], **Hinata SEGAWA**[1], and **Tetsuro KITAHARA** (kitahara@kthrlab.jp)[1]

[1]**Nihon University**, Tokyo, Japan

ABSTRACT

In the finger-style guitar, players often play both melody notes and chords simultaneously with a single guitar. This playing style is difficult because the player has to find a playable set of fretboard positions that include both melody notes and chords. In this paper, we propose a method for generating a tablature that enables users to play both melody notes and chords simultaneously from a given lead sheet, which consists of melody and chord symbol notations. Our method achieves the tablature generation as an optimaization problem of searching for the minimal-cost sequence of states. A combination of the fingering positions for the five fingers on the fretboard is regarded as a state. And three types of performing costs (an initial cost, a transition cost, and an emission cost) are formulated for each state. Then, a sequence of states having the minimal cost is searched with a Viterbi algorithm. Experimental results showed that tablatures generated with our method were moderately evaluated by an expert of the classical guitar.

1. INTRODUCTION

The finger-style guitar is one of the common playing styles for acoustic guitars. Unlike plectrum playing, the finger-style guitar enables guitarists to play complicated performances because they can independently pluck different strings with different fingers. Indeed, it is common to play both melody notes and chords simultaneously with a single guitar without any accompaning players. This playing style is called *solo guitar* in this paper.

The solo guitar is considered a difficult playing style. One of the major difficulties resides in the fact that it is complicated to find a playable set of fingering positions on the fretbaord, which play both melody notes and chords. Because melody notes, chord notes, and bass notes are played with a single guitar, the fingering positions of all fingers for the melody, chord, and bass notes have to fall within a physically playable range. Searching for such fingering positions is not easy for many non-professional players.

The purpose of this study is to achieve automatic generation of a tablature for the finger-style solo guitar for a given lead sheet. Because a lead sheet consists of the notations of melody notes and chord symbols, there are multiple possibilities of the voicing for each chord. To generate a playable tablature, the system has to carefully choose an appropriate voicing that can be played together with the melody notes.

Automatic tablature generation has been attempted by a number of researchers, but no one has attempted tablature generation from a lead sheet under the assumption that both the melody and the chord progression are played with a single guitar. In 2004, Miura et al. developed a system that selects the optimal fingering positions for monophonic melodies by minimizing finger motions on the fretboard [1]. In 2005, Tuohy et al. applied a genetic algorithm to find playable tablatures by using a fitness function that assesses the playability of a given set of fretboard positions. They used simple polyphonic music in their experiment but the inputs were not lead sheets [2]. In 2016, Hori et al. proposed a minimax Viterbi algorithm to minimize the maximal difficulty in fingering motion on the fretboard. They tested their method with simple monophonic melodies [3]. In addition, there have been some attempts to introduce a tablature representation in automatic music transcription to consider the playability in the guitar (e.g., [4]). Their inputs were audio signals, not lead sheets.

In our previous paper [5], we developed a tablature generation system for polyphony that consists of only melodies and bass lines. Because the number of simultaneously played notes are one or two in such polyphony, we defined a set of states that represents fingering positions for one or two simultaneous notes. The system then generates a tablature by searching for the state sequence that has the minimal performing cost.

To extend this method to a combination of melody notes and chords, we add states that represent more than two notes. However, if this addition is simply done, we will have two problems: (1) states that represent unsuitable (very difficult-to-play) note combinations may be added, and (2) the computational cost may become higher due to the increase of the number of states. To avoid these problems, we introduce *typical forms*, a set of very common fingering forms for commonly used chords. By introducing them, we limit the set of states to common fingering forms, so we can avoid that unsuitable fingering forms are contained in generated tablatures.

2. TABLATURE GENERATION ALGORITHM

Our system generates a tablature from a lead sheet consisting of a melody and a chord progression. Because

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

our target is finger-style solo guitar, in which the guitarist plays both melody notes and chords simultaneously, fingering (string-pressing) positions on the fretboard for melody notes and chords must fall within a physically playable range. We achieve this by optimizing state transitions in which physically unplayable states had been removed.

2.1 Data representation

We denote a gjven melody and chord progression as $X = \{(x_1, c_1), \cdots, (x_N, c_N)\}$, where x_n is the MIDI note number (or rest) of the n-th note and c_n is the chord symbol assigned to the n-th note. In the current implementation, each chord is played only at its beginning time; for other timings (no chord symbol or a chord continues from the previous note), a special symbol ϵ is set to c_n.

Next, we define a 6-dimensional vector, $\boldsymbol{v} = (v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}, v^{(6)})$, which represents a fingering form on the fretboard. $v^{(m)}$ represents the fret position of the finger pressing the m-th string ($v^{(m)} = 1, \cdots, 14$). $v^{(m)} = 0$ represents an open string (the string is not pressed although plucked) and $v^{(m)} = -1$ represents no plucking.

Let V be a playable set of fingering form vectors $\boldsymbol{v}$. When we denote the optimal fingering form for (x_n, c_n) as $\boldsymbol{q}_n$ ($\in V$), our optimization problem can be formulated as:

$$\text{minimize } C(Q) \text{ s.t. } Q = \{\boldsymbol{q}_1, \cdots, \boldsymbol{q}_N\}$$

where $C(Q)$ is a performing cost function (representing the degree of the non-optimality).

The cost function consists of the following three types of costs:

- **Initial cost** $C(\boldsymbol{q}_1)$

  It is said that neck-side positions are more common than body-side positions in classical guitar, so lower costs are set to neck-side positions.

- **Transition cost** $C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n)$

  As the motion of the fingering positions from $\boldsymbol{q}_n$ to $\boldsymbol{q}_{n+1}$ is smaller, playing it becomes easier. Therefore, we give lower costs to smaller motions of the fingering positions.

- **Emission cost** $C((x_n, c_n)|\boldsymbol{q}_n)$

  The fingering form $\boldsymbol{q}_n$ must emit the melody note x_n and chord tones of the chord c_n. In addition, it is more desirable to emit more chord tones to make a rich harmony. The emission cost is defined from these points of view.

The overall cost $C(Q)$ is defined by the following equation:

$$\begin{aligned} C(Q) &= C(\boldsymbol{q}_1) \\ &\quad + \sum_{n=1}^{N-1} \left\{ C((x_n, c_n)|\boldsymbol{q}_n) + C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n) \right\} \\ &\quad + C((x_N, c_N)|\boldsymbol{q}_N) \end{aligned}$$

This formulation is inspired by a hidden Markov model.

2.2 Defining a playable set of fingering form vectors

As mentioned above, V is a playable set of fingering form vectors. Each element $\boldsymbol{v} \in V$ is a 6-dimensional vector $\boldsymbol{v} = (v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}, v^{(6)})$, where $v^{(m)}$ represents the fret position of the finger pressing the m-th string.

To define V, we adopt the following policies for chord voicing:

- The highest note shall always be a melody note.
- The lowest note (bass note) shall always be the root note.
- The fretboard positions for different strings should be close enough to each other (within three frets).
- Fingering forms familiar for guitarists are preferred.

We add fingering position vectors to V based on the following methods.

2.2.1 The case of two or less simultaneous note(s)

When the number of simultaneous notes of a fingering position vector is two or less, this fingering is used to play only a melody note or a combination of a melody note and a bass note. We therefore add fingering form vectors $\boldsymbol{v}$ satisfying the following requirements to V:

1. $\max_+(\boldsymbol{v}) - \min_+(\boldsymbol{v}) \leq 3$
2. $n(\{m \mid v^{(m)} > 0\}) < 2$.
3. Multiple strings do not correspond to the same note

where $\max_+(\boldsymbol{v})$ and $\min_+(\boldsymbol{v})$ denote the maximum and minimum values of the positive-value elements of $\boldsymbol{v}$, respectively. $n(\cdot)$ is the number of the elements in a given set.

2.2.2 The case of three or more simultaneous notes

When the number of simultaneous notes is three or more, the fingering is used to play both a melody note and chords simultaneously (or chord only if the melody note is a rest). The important here is to give a higher priority to more familiar fingering forms for players. To achieve this, we define *typical forms*, which represent fingering forms for basic chords for familiar for guitarists. For example, the typical forms for the chords C and F are $(0, 1, 0, 2, 3, -1)$ and $(1, 1, 2, 3, 3, 1)$, respectively. We manually made a dataset that include 3658 typical forms.

In addition to all of the 3658 typical forms, *modified forms* are added to V. The modified forms are fingering forms in which a few elements are changed from a typical form. They were made by changing each typical form $\boldsymbol{v}$ as follows:

- Reduce string-pressing:
  Replace one or two element(s) of $\boldsymbol{v}$ with 0 (an open string) or -1 (no plucking).
- Add string-pressing:
  Replace one of the elements having 0 or -1 with the maximal value of $\boldsymbol{v}$, if the little finger is not

used. For example, the chord C, the fingering form of which is $(0, 1, 0, 2, 3, -1)$, is played without the little finger. Then, modified forms $(3, 1, 0, 2, 3, -1)$ and $(0, 1, 3, 2, 3, -1)$ are generated.

2.3 Narrowing down the set of fingering form vectors

The set of fingering form vectors, V, have a large number of vectors, so it requires a high computational cost to directly search the optimal fingering form from V. Therefore, we narrow down V. For n-th note (x_n, c_n), the elements that do not emit the melody note x_n and the chord c_n are removed from V. The narrowed-down set for (x_n, c_n) is denoted as V_n. Specifically, V_n has only vectors $\boldsymbol{v} = (v^{(1)}, v^{(2)}, v^{(3)}, v^{(4)}, v^{(5)}, v^{(6)})$ that satisfy the following:

- The highest note matches the melody note.
  $\exists m_0 : \mathrm{note}(v^{(m_0)}) = x_n$ and $\forall m < m_0 : v^{(m)} = -1$.
- The lowest note matches the root note of the chord.
  $\exists m_1 : \mathrm{note}(v^{(m_1)}) = \mathrm{root}(c_n)$ and $\forall m > m_1 : v^{(m)} = -1$.
- The notes between the highest and lowest notes are chord notes.
  $\forall m \in (m_0, m_1) : \mathrm{note}(v^{(m)}) \in \mathrm{notes}(c_n)$.

$\mathrm{note}(v^{(m)})$ denotes the note corresponding to the fret position $v^{(m)}$ for string m, $\mathrm{root}(c_n)$ denotes the root note of the chord c_n, and $\mathrm{notes}(c_n)$ denotes the set of the chord notes of c_n.

When $c_n = \epsilon$, only the melody note is played. Therefore, V_n only includes vectors that have a value satisfying $\mathrm{notes}(v^{(m)}) = x_n$ for one element m and -1 for the others.

2.4 Defining initial costs

On acoustic guitars, positions closer to the neck is more common than positions closer to the body, so positions closer to the body should have a slightly higher cost.

$$C(\boldsymbol{q}_1) = \begin{cases} 2.5 & (\max_+(\boldsymbol{q}_1) \leq 4) \\ 5.0 & (\text{otherwise}) \end{cases}$$

2.5 Defining transition costs

We define the transition cost to reduce the difficulty of the performance by increasing the cost of state transitions that involve large motions. In addition, for the same reason as above, a higher priority is given to positions closer to the neck.

$$C(\boldsymbol{q}_{n+1}|\boldsymbol{q}_n) = \begin{cases} 0.0 & (\max(\boldsymbol{q}_{n+1}) = 0 \text{ or } \max(\boldsymbol{q}_n) = 0) \\ |\min_+(\boldsymbol{q}_n) - \min_+(\boldsymbol{q}_{n+1})| + \alpha & (\text{otherwise}) \end{cases}$$

The α is introduced to give a higher preference to fingering positions closer to the neck (5.0 when $\max(\boldsymbol{q}_{n+1}) \geq 5$; 0.0 otherwise).

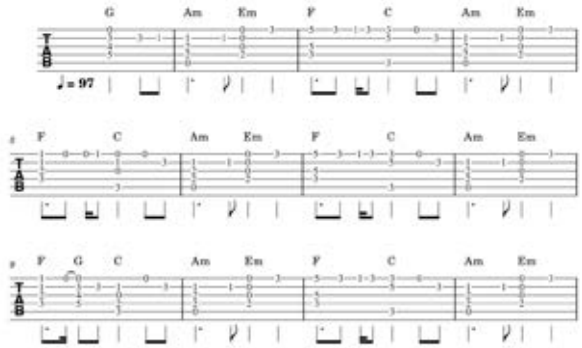


Figure 1. An example of the tabulatures generated by our system

2.6 Definng emission costs

Because the vectors that do not emit (x_n, c_n) have been removed from V_n, we do not have to consider if the fingering vector emits the melody note x_n and the chord c_n when we define the emission cost. We therefore define the emission cost based on the following policy:

- When the fingering vector is not modified from a typical form, a lower emission cost is given, because typical forms are expected to be easier to play than modified forms.
- For fingering vectors that emits more tones, a lower emission cost is given, because such vectors are expected to make a richer harmony.

Specifically, we define the emission cost as follows:

(i) When $c_n = \epsilon$ We simply set 0 to $C((x_n, c_n)|\boldsymbol{q}_n)$.

(ii) When $c_n \neq \epsilon$ We define the emission cost as:

$$C((x_n, c_n)|\boldsymbol{q}_n) = \mathrm{voices}(\boldsymbol{q}_n) + \mathrm{typical}(\boldsymbol{q}_n),$$

where $\mathrm{voices}(\boldsymbol{q}_n)$ is a cost calculated from the number of simultaneous notes and $\mathrm{typical}(\boldsymbol{q}_n)$ is a cost determined from whether the fingering form is a typical form. In defining $\mathrm{voices}(\boldsymbol{q}_n)$, we consider the number of simultanously plucked strings $\mathrm{pluck}(\boldsymbol{q}_\mathrm{n})$:

$$\mathrm{voices}(\boldsymbol{q}_n) = \begin{cases} \infty & (\mathrm{pluck}(\boldsymbol{q}_n) = 6) \\ \infty & (\mathrm{pluck}(\boldsymbol{q}_n) = 5 \\ & \text{and } \mathrm{dur}(x_n) < 2) \\ -\beta\ \mathrm{pluck}(\boldsymbol{q}_n) & (\text{otherwise}) \end{cases}$$

where $\mathrm{dur}(x_n)$ is the duration of the melody note x_n ($\mathrm{dur}(x_n) < 2$ means that x_n is shorter than a half note) and β is a positive constant. $\mathrm{typical}(\boldsymbol{q}_n)$ is defined as:

$$\mathrm{typical}(\boldsymbol{q}_n) = \begin{cases} 10 & (\boldsymbol{q}_n \text{ is a typical form}) \\ 12 & (\text{otherwise}) \end{cases}$$

3. EXPERIMENT

3.1 Method

The tablatures of 8 pieces of music output by the proposed method (an example is shown in Fig. 1) were evaluated by a classical guitar expert. For each piece, she answered the questions in table 1 on a 7-point scale.

Table 1. Questions asked to the evaluator

Q1	Was there anything in the voicing of each chord that made it difficult to play?
Q2	Did you find the transition from chord to chord difficult to play?
Q3	Did you find any part of the melody difficult to play?
Q4	Do you think the voicing is musically appropriate?
Q5	Was the difficulty level appropriate for intermediate level players?
Q6	Do you think the arrangement is useful as a starting point for intermediate players to make their own arrangements?

Table 2. Evaluation of each piece

	Q1	Q2	Q3	Q4	Q5	Q6
Let it Snow	6	4	4	3	5	4
The Wellerman	5	5	5	6	5	5
Canary	7	6	7	5	7	5
Cherry	7	7	6	6	7	7
Sora mo Toberu hazu	6	4	6	5	6	6
Tsuki wo Miteita	6	5	6	5	7	6
Uchiage Hanabi	5	6	7	5	7	6
Yoake to Hotaru	4	3	4	5	5	5

3.2 Results

The results are listed in Table 2. From the table, we can see that the overall evaluation was good. However, there were some large differences in evaluation depending on the music piece, such as "Cherry" and "Yoake to Hotaru".

For "Let it Snow", the ratings for Q2, Q3, Q4, and Q6 were low. Regarding Q4, we received the opnion that the voicings (especially the number of simultaneous notes) should be changed based on the metrical position; the harmony should be richer for the begining of each measure and less richer for the rest). For "Sora mo Toberu hazu", the rating for Q2 was low. This is because the number of simultaneous notes were too large, caused by the definition of the emission cost.

4. CONCLUSION

In this paper, we proposed a tablature generation system for solo guitar, in which the melody and chords are played simultaneously. To obtain the optimal fingering form for a given melody and chord notes, the Viterbi algorithm is applied with three types of costs: initial costs, transition costs, and emission costs. Through the experiment, we confirmed the the generated tablatures are moderaltely good, but they have parts that are difficult to play, especially for chord voicings. Finding a tradeoff between ease-to-play and harmonic richness is one of the important remaining issues.

Acknowledgments

This work was supported by KAKENHI Grants-in-Aid for Scientific Research 22H03711 and 21H03572. We thank Dr. Nami Iino for her help in the experiment.

5. REFERENCES

[1] M. Miura, I. Hirota, N. Hama, and M. Yanagida: Constructing a system for finger• position determination and tablature generation for playing melodies on guitars, Systems and Computers in Japan, vol. 35, no. 6, pp. 10–19, 2004.

[2] D.R.Tuohy and W.D.Potter: A Genetic Algolithm for the Automatic Generation of Playable Guitar Tablature", Proceedings of ICMC 2005, pp. 499–502, 2005.

[3] G. Hori and S. Sagayama: Minimax Viterbi algorithm for HMM-based guitar fingering decision, Proceedings of ISMIR 2016, pp. 448–453, 2016.

[4] K. Yazawa, D. Sakaue, K. Nagira, K. Itoyama, and H. G. Okuno: Audio-based guitar tablature transcription using multipitch analysis and playability constraints, Proceedings of ICASSP 2013, pp. 196–200, 2013.

[5] S. Sakai, H. Segawa and T. Kitahara: Generating Tablature of Polyphony Consisting of Melody and Bass Line, Proceedings of CMMR 2023, pp. 724–727, 2023.

EXPLORING UNCHARTED SOUNDSCAPES: INNOVATIVE INTERACTION AND MAPPING TECHNIQUES WITH BODYHARP AND THE TEINOPHON

Leo FOGADIĆ[1], **Doga Buse CAVDIR**[2], and **Dan OVERHOLT**[3]

[1,2,3]**Aalborg University**, Copenhagen, Denmark

ABSTRACT

The evolution of Digital Musical Instruments (DMIs) has pushed the boundaries of musical expression and interaction. These instruments leverage advanced technologies to offer new dimensions of creative possibilities. This paper investigates the musical possibilities of Bodyharp [1] and the Teinophon [2], each designed to go beyond the limits of traditional instruments. We aim to uncover new perspectives on performer interaction and mapping, using the capabilities of these instruments to surpass traditional stringed-instrument boundaries with their infusion of sensor-based technologies.

The study focuses on crafting and analyzing inventive mappings that connect the performer's gestures to sound, extending digital musical interactions beyond traditional instrument techniques dictated by physical and acoustical limitations. We explore uncharted sonic territories, drawing inspiration from the instruments' familiarities, yet extending them to create unique musical interactions.

Through qualitative and performance-based examination of the interaction between body movements and sonic outcomes, we identify mappings that offer the most compelling and engaging results. This research recognizes the personal nature of musical preferences while encouraging creative exploration following a workshop-style evaluation method. We collected data from eight participants based on their experience with exploring both instruments' affordances and with solo and collaborative performance practices.

Our work contributes to the broader discussion on sensor-based instruments, providing insights into their potential for expanding musical expression beyond established norms. By sparking further innovation and exploration, we hope to deepen our understanding of the creative possibilities embedded in Bodyharp and Teinophon, thus paving the way for new dimensions in contemporary musical expression.

1. INTRODUCTION

In recent years, the intersection of musical instruments and sensor-based technologies has given rise to a new realm of possibilities in musical expression. This study explores two similar yet distinctive instruments, Bodyharp and the Teinophon (see Figure 1), both designed to exceed the conventional boundaries of traditional string instruments through innovative interaction and mapping techniques [1] .The real-time sensor technology in these instruments opens avenues for performers to engage with sound in unprecedented ways, breaking away from established practices.

As sensor technologies advance, so does the potential for creating instruments that respond intimately to the performer's gestures. This builds upon existing research in the field of Digital Musical Instruments (DMI), emphasizing the tools we can borrow from Human-Computer Interaction (HCI) [3]. Furthermore, our study aligns with the broader discourse on musical gestures and their correlation with sound, as explored by Godøy and Leman [4], and the conceptual framework of 4E cognition and dynamical systems theory proposed by van der Schyff [5].

Figure 1. BodyHarp and Teinophon co-performance

Through this investigation, we seek to answer the question: "How can Bodyharp and the Teinophon facilitate novel interaction and mapping techniques, transcending the constraints of traditional string instruments, and which mappings yield the most musically compelling outcomes?" By addressing these things, we aim to uncover fresh perspectives on performer interaction, offering insights into the expressive possibilities embedded in these instruments.

[1] A performance excerpt of the two instruments collaboration can be accessed here: `https://youtu.be/lDolhml1xfg`

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

The objectives of the research are:

- Exploring inventive mappings connecting performer gestures to sonic outputs on Bodyharp and the Teinophon.
- Identifying mappings that yield the most compelling and engaging musical outcomes.
- Contributing to the broader discussion on sensor-based instruments, expanding the understanding of musical expression with DMIs.

2. BACKGROUND AND RELATED WORK

In his research around *action-sound* couplings and relationships, Jensenius draws attention to how artificial *action-sound* relationships will never be as solid as an *action-sound* couplings [6]. However, in a scenario when designing DMIs and testing them with various types of sound engines, Jensenius describes it as a potentially powerful way of exploring action-sound relationships in practice. This approach may provide valuable insights into some of the underlying features of our perception of action-sound relationships.

In addition, we take inspiration from Overholt's discussions concerning how best to leverage certain properties of acoustic instruments and synthesis algorithms, introducing them into the development of DMIs [7]. Finding mappings that excel at translating a performer's gestures into sound in ways that can evoke the intended affective qualities, while maintaining each instrument's musical identity is a complex process. The many types of effort involved are nicely outlined in Baalman's book on mapping, Composing Interactions [8].

3. INSTRUMENTS OVERVIEW

Bodyharp, initially developed in 2018 [1], is a semi-wearable instrument, consisting of an enclosure where strings are placed, and a wearable arm piece [1]. The iteration used in this study, shown in Figure 2, features a redesigned 3D-printed enclosure and a wearable hand controller [9]. The hand controller serves as an interface with digital sensors, such as buttons, force-sensing resistors, and an accelerometer, allowing the musician to have more nuanced control over the sound parameters. Additionally, the controller houses the main controller, a Teensy board [2], which handles all the input data and sends it to a computer for further processing.

Bodyharp's sound mapping follows a gesture-based mapping model [10] on two scales of gestures. The larger-scale gestures contribute to sound production (through plucking or stretching the strings) while small-scale, nuanced gestures control the sound effects with the following mapping strategy: (1) The push buttons on the hand controller change the chord progression in three scales; (2) The pressure (or force-sensitive resistor, FSR) sensor, positioned under the thumb, controls the quality of the filter, creating a tremolo effect with the dabbing gesture; (3) One slider controls the gain and the other one controls the note duration by changing the time constant of the string model; (4) The square pressure sensor, positioned on the back of the hand controller, increases the drive of the filter. With a delay coupled with this effect, touch interaction creates an echo effect.

Figure 2. Bodyharp

Figure 3. The Teinophon

The Teinophon, firstly constructed in 2021 [2], is a tabletop instrument with a simple interface consisting of 7 horizontally laid strings parallel to one another. The iteration used in this study, shown in Figure 3, is an improved version of the Teinophon. It is built with a more durable custom-made wooden enclosure. Furthermore, the spring mechanism uses a different, more robust design. Lastly, this latest iteration of the Teinophon features a Bela board [3] for processing input signals and synthesizing sound outputs.

Teinophon's sound mapping also follows a gesture-based mapping model, focusing on the interaction between the performer and the strings. One mapping involves the detection of plucking events, where the velocity and intensity of the sound are directly influenced by the displacement

[2] https://www.pjrc.com/teensy/

[3] https://bela.io

and release speed of the strings. Furthermore, the instrument maps the lateral movements and pressure exerted on the strings to control parameters bow velocity and bow position. Additionally, low-pass filters are applied and their cutoff frequency is mapped to the height of each pulled string. The interactions are processed in real-time, allowing for dynamic and expressive performances. These intuitive mappings enables performers to produce a wide range of sounds by varying their gestures and interactions with the strings.

4. METHODOLOGY

4.1 Participants

Eight participants attended the workshop, ranging in age from 26 to 52 years. Although participants had different levels of musical experiences, a background in music was not required for this study. All participants were informed about the nature of the study and signed our consent form before participating.

4.2 Instruments

The instruments were introduced to participants at the beginning of each session. Detailed explanations of each instrument's components, playing techniques, and sonic possibilities were provided to ensure participants had a foundational understanding before engaging in the evaluation sessions.

4.3 Evaluation Sessions

The study employed a mixed-methods approach, combining quantitative data collection through Likert-style questionnaires with qualitative insights gathered during post-performance discussions. The participants' interactions were observed by the researchers in both solo and co-performance sessions.

4.3.1 Solo Performances

Each participant was given dedicated time for solo exploration with both Bodyharp and the Teinophon. During this phase, participants were encouraged to experiment with various playing techniques, such as plucking, pulling, and other techniques such as transverse string displacements. The goal was to allow participants to familiarize themselves with the instruments and uncover potential nuances in sonic expression.

4.3.2 Impressions And Questions

Participants were given a chance to provide short feedback about the two instruments, and get more detailed information, about playing techniques.

4.3.3 Co-Performance

Following the solo sessions, participants engaged in collaborative performances in pairs and improvised Bodyharp/Teinophon duets. This segment of the mapping workshop aimed to investigate how the instruments interacted in a shared musical space, exploring potential synergies and challenges in combining Bodyharp and Teinophon. The co-performance sessions were recorded for further analysis, and all participants were given a post-workshop interview as a chance to provide any final reflections.

4.4 Data Collection

Data collection occurred through a two-phase process:

4.4.1 Questionnaires

Structured questionnaires were designed to capture participants' subjective experiences, preferences, and challenges with each instrument. The questions assessed factors such as ease of use, expressiveness, and overall satisfaction [11]. Open-ended questions allowed participants to provide detailed qualitative feedback.

4.4.2 Post-Performance Discussions

The post-performance discussion was conducted in pairs. This qualitative phase aimed to delve deeper into participants' experiences, uncovering insights that might not be captured by quantitative measures. Participants were encouraged to share preferences, and any notable challenges encountered during the performances.

5. MAPPING TECHNIQUES

Besides the sound synthesis engine and the control interface, mapping is one of the vital aspects of designing digital musical instruments. With endless possibilities, it can be a fun but challenging task to implement mappings which will make the instrument expressive and intuitive to use. Common techniques in mapping include parameter scaling, non-linear functions, and convergent mappings.

In a study of a mapping design process, West et al. point out that effective mappings should consider the balance of musical agency between the player and the instrument, primarily empowering the player to perform specific sounds as they intend to, but perhaps sometimes allowing the instrument to behave unexpectedly [12].

The practice of mapping dates back as far as the inception of acoustic instruments themselves. However, it is only with the development of real-time electronic instruments that designers have actively integrated flexible mappings into each instrument. It is proven that mappings are more effective at eliciting a good performance from a human player when the performer is confronted with multi-parametric tasks, more so than a series of one-to-one mappings [13].

The mappings implemented for the workshop were both loosely inspired by traditional string-instruments, slightly more directly on the Teinophon than on BodyHarp. Synthesis algorithms used on the Teinophon include both plucked (Karplus-Strong) and bowed (Waveguide) string physical models, written in C++ on Bela; mappings involve event detection for plucks, amount of pulling the string for bow pressure and velocity parameters, and detection of transverse string displacement for determining the bowing position parameters of each string. For BodyHarp, complex mappings strategies, using Faust physical

models that are implemented in ChucK audio programming language, were utilized. The mappings include layers of one-to-one mapping, such as mapping string height to chords with higher pitch classes, to many-to-many parameters, such as mapping force sensitive resistor (FSR) and accelerometer data to sound synthesis algorithms parameters (for a detailed review on sound design, please see [9]).

6. EVALUATION RESULTS

6.1 Participant Experience

6.1.1 Instrument Familiarity and Musicality

Participants' evaluations shed light on their perceptions of the instruments' familiarity and musicality, crucial aspects in understanding their engagement with Bodyharp and the Teinophon. Bodyharp's more complex mappings presented challenges in translating gestural input into expressive musical output. On the other hand, participants deemed the Teinophon more intuitive to play, attributing it higher musicality ratings. This distinction in perceived familiarity could influence participants' comfort levels and creativity with each instrument.

6.1.2 Control and Intuitive Interaction

Control dynamics and the perceived naturalness of interactions played pivotal roles in participants' evaluations of the instruments.

Participants reported a moderate sense of control over Bodyharp, and minor concerns were raised regarding the control of the produced sounds with the gestural interactions. The more traditional-looking string interface of the Teinophon seemed to resonate positively with participants, contributing to a more transparent experience. However, no conclusions can be drawn from this due to the lack of any longitudinal evaluations.

6.2 Sound Preferences and Creativity

Positive feedback was expressed on the sound qualities of the instruments. The participants evaluated the physical models as sounding good, and detailed. However, some of them have expressed that they would enjoy hearing completely different and unexpected sounds.

Feedback on Bodyharp indicated positive responses to the produced sounds. Participants reported feeling creatively inspired, exploring unconventional sonic outcomes. However, there were suggestions for expanding the variety of sounds.

Similar positive sentiments were expressed regarding the Teinophon, with participants appreciating the instrument's capacity to evoke creative exploration. However, similar to Bodyharp, participants suggested more diverse sonic outputs.

6.2.1 Collaborative Performance Feedback

Participants' reflections on collaborative performances provided insights into the instruments' synergy and their impact on shared musical spaces.

Collaborative performances yielded generally positive feedback, with participants navigating the challenge of combining the unique sonic characteristics of both instruments.

Furthermore, the participants acknowledged the complementarity of the instruments and expressed enjoying how the two instruments quickly went from being harmonious to sounding dissonant. These moments evoked certain emotions with the participants. They described that during the performances, their emotions ranged from peaceful to *spooky*.

Additionally, the participants expressed positive remarks on the joint movement efforts. Many of them noted how they were observing the other performer and trying to coordinate their movements. One participant phrased the experience as: "...showing people sound ... almost like a dance".

Lastly, instances of sonic overlaps were noted during collaborative sessions, signaling potential areas for adjusting mappings to maintain clarity and coherence in joint performances.

6.3 Mapping Feedback

Participants provided nuanced insights into their experiences with the instrument mappings, addressing specific aspects related to the gestural mappings employed for sonic expression.

6.3.1 Bodyharp Mappings

Overall, positive feedback was received for the mappings. The participants expressed their appreciation of using their whole bodies to play the instruments. One participant even compared their movements to the martial arts *Tai Chi*. However, the test subjects suggested clearer representations of gesture-to-sound relationships. Participants emphasized the importance of understanding which sounds were being controlled to enhance their expressive capabilities.

6.3.2 Teinophon Mappings

Participants appreciated the coherence between gestural input and sonic output. Generally, positive feedback was received for the mappings. Specific suggestions were made for Teinophon, to refine plucking control with adding a *mute* feature when touching the strings in their resting position, and plucking one side of the strings while they are raised. Other suggestions were made such as adding higher level mappings, for example detecting the acceleration and jerk of movements to control the sound output. Additionally, the participants suggested adding effects like vibrato.

7. DISCUSSION

7.1 Familiarity and Musicality

7.1.1 Bodyharp's Unique Affordances

Bodyharp's innovative gestural interactions presented a distinctive way for performers to engage with music. When the performer is embodied with the instrument it

opens up new possibilities for creative exploration. Future improvements should aim to focus even more on these unique affordances, providing clearer mappings to enhance performers' understanding of the instrument and its sonic capabilities.

7.1.2 Teinophon's Intuitive Design

Teinophon's more traditional looking string interface was perceived as familiar, reminiscent of a harp or a guitar, contributing to an intuitive playing experience. This feedback suggest incorporating familiar elements into novel digital musical instruments, facilitating a smoother learning experience. Future iterations of Teinophon should keep a good balance of familiarity and innovative interaction.

7.2 Sonic Preferences and Creativity

7.2.1 Diverse Sonic Palette

Even though an appreciation of both instruments' somewhat traditional sounding qualities was expressed, the participants would have liked a more diverse sonic palette and hearing unexpected sounds. Future enhancements could potentially facilitate more variety of sounds with different synthesis algorithms.

7.2.2 Collaborative Performance Dynamics

In the collaborative performances, the instruments demonstrated a capability to induce emotional responses. This positive feedback highlights the instruments' suitability for collaborative contexts. It was shown that playing these instruments together, the performers compliment each other and are aware of their joint performance through sonic and visual feedback.

7.3 Mapping Feedback

7.3.1 Bodyharp's Expressive Mappings

Participants have expressed appreciation for Bodyharp's mappings and praised the engagement of the whole body when playing. The comparison to *Tai Chi* underscores the instrument's holistic and embodied experience. Future iterations should focus on refining mappings to provide clearer representations, empowering performers to fully harness the expressive capabilities inherent in Bodyharp.

7.3.2 Teinophon's Successful Translation

The instrument's well-received mappings proved a successful translation of gestural input to sonic output. Participants valued the coherence between actions and sounds. Improvements, such as the suggested mute feature and higher-level mappings, open paths for further innovation, maintaining the instrument's positive trajectory in delivering intuitive and expressive mappings.

7.4 Practice and Time Investment

Since each session lasted around 1 hour and 30 minutes per pair, participants expressed a shared desire for extended practice time. This would have allowed them to get more comfortable with the instruments. This was a collective opinion that emphasized the need for future studies to consider longer practice times. This would impact performers' comfort levels and creative exploration.

8. CONCLUSION

Our study emphasizes the positive aspects of both Bodyharp and the Teinophon while providing constructive feedback for future development.

Bodyharp, with its unique affordances and gestural interaction, presents an excellent example of embodied interaction and a holistic approach to performances. Through our study, participants highlighted the instrument's capacity to induce creative exploration. Despite the suggestion for clearer mappings, the overall opinion leaned towards a positive experience in solo and/or collaborative performance.

Turning the focus to the Teinophon, our study has shed light on its intuitive interface, providing the performers with a familiar yet technologically enhanced string interface. Participants complimented the transparency it offers, emphasizing the ease of play and instant musicality. Suggestions for more refined plucking mappings and incorporating new mappings open the door for future developments.

Acknowledgments

We would like to thank all the participants who attended the workshop and contributed to our research. We are also thankful for their wonderful performances.

9. REFERENCES

[1] D. B. Cavdir, R. Michon, and G. Wang, "The bodyharp: Designing the intersection between the instrument and the body," in *Proceedings of the 15th Sound and Music Computing Conference*, 2018, pp. 498–503.

[2] L. Fogadić and D. Overholt, "The Teinophon," in *Proceedings of the 19th Sound and Music Computing Conference*, 2022, pp. 442–445.

[3] N. Orio, N. Schnell, and M. Wanderley, "Input Devices for Musical Expression: Borrowing Tools from HCI," 01 2001.

[4] R. Godøy and M. Leman, "Musical Gestures: Sound, Movement, and Meaning," 01 2009.

[5] D. van der Schyff, A. Schiavio, A. Walton, V. Velardo, and A. Chemero, "Musical creativity and the embodied mind: Exploring the possibilities of 4E cognition and dynamical systems theory," vol. 1, pp. 1–18, 09 2018.

[6] A. Jensenius, "ACTION–SOUND - Developing Methods and Tools to Study Music-Related Body Movement," Ph.D. dissertation, 01 2007.

[7] D. Overholt, "Designing Interactive Musical Interfaces," in *Designing Interactions for Music and Sound*, M. Filimowicz, Ed. Focal Press, 2022, pp. 1–29.

[8] M. Baalman, *Composing Interactions - An Artist's Guide to Building Expressive Interactive Systems*. V2 Publishing, 2022.

[9] D. Cavdir and G. Wang, "Designing felt experiences with movement-based, wearable musical instruments: From inclusive practices toward participatory design," *Wearable Technologies*, vol. 3, p. e19, 2022.

[10] Cavdir, Doga and Wang, Ge, "Borrowed Gestures: The Body as an Extension of the Musical Instrument," *Computer Music Journal*, vol. 45, no. 3, pp. 58–80, 09 2021. [Online]. Available: https://doi.org/10.1162/comj_a_00617

[11] G.-M. Schmid, *Evaluating the Experiential Quality of Musical Instruments*. Springer, 2017.

[12] T. West, B. Caramiaux, S. Huot, and M. Wanderley, "Making Mappings: Design Criteria for Live Performance," in *International Conference on New Interfaces for Musical Expression*, 06 2021.

[13] A. Hunt and M. Wanderley, "Mapping performer parameters to synthesis engines," *Organised Sound*, vol. 7, pp. 97 – 108, 08 2002.

PROGRAM NOTES

CONCERT 1

/ Terminal

/ Travelling without moving

/ Too waste, too fast

/ Matters 8

TERMINAL

Leonie Strecker
University of Music and Performing Arts Graz

Terminal is a composition for one performer, live-electronics and multichannel-tape. The performer uses her voice to control spatialization and sound synthesis parameters of an immersive audio landscape that itself consists of different layers of synthesized and recorded voices, while her own voice is inaudible to the audience. The fact that her voice exerts control while being inaudible invites reflection on gradations and shadings of power in an audio landscape that explores issues of power, control, and self-determination. The sound material in this piece is to a large extend derived from interviews that the composer recorded with people that are close to her in different ways. Most importantly, the voices of her mother and her former composition teacher are heard. In reflecting her relation to these voices and the impact they had on the composer, she examines the structures of power that can be expressed through the voice in a personal narration.
Terminal opens up a speculative, immersive space in which different voices, captured at different times and in different locations, synthesized voices and crowds meet and merge. The sound of the disembodied voices is guided by the performer that is present, yet inaudible. Connections between these imagined and present bodies form and disintegrate, emerge, dissolve, creating layers of meaning and interdependence.
The piece was commissioned by La Biennale di Venezia and premiered in Venice in October 2023 for the La Biennale di Venezia Micro-Music Edition. It was further developed within the Speculative Sound Synthesis project. Speculative Sound Synthesis is an Artistic Research project hosted at the Institute for Electronic Music and Acoustics of the University of Music and Performing Arts Graz. It is funded by the Austrian Science Fund within the program for arts-based research (PEEK) – PEEK AR 713-G.

TRAVELLING WITHOUT MOVING

Stefano Catena
De Montfort University

“Travelling without moving” (8’10”) is a journey in acousmatic composed spaces and environments, from granular rainy textures to sonic trajectories circling around the listener. The concepts of “journey” and “motion” are investigated musically: the experience “moves” between soundscapes, leading to always-changing sonic worlds, both from the spectral and spatial perspectives. The piece was composed during a residency at the Royal College of Music in Stockholm, and most of the atmospheres are inspired by the acoustic experiences I had during my stay in the city. More importantly, the spatial component of the music is paramount, as the piece was composed for the 29.4 partial dome system hosted at KMH in Lilla Salen. The piece was inspired by natural soundscapes of Stockholm, where it was written, but it is also influenced by the nostalgia of a distant home: this is where the mind travels, even though its not really moving.

TOO WASTE, TOO FAST

Francesco Perissi

"Too waste, too fast" (2024 - 12"32') is a composition part of "X6 - Hexaphonic Augmented Guitar project" in which a special divided pick-up enables the separation of the 6 strings channels. The project is based on an instrument played live that allows to control not only dynamics and frequency but also the acoustic space. In this case all the strings are connected to different speakers building a 6.1 surround audio system or wide stereo configuration or binaural listening.
"Too waste, too fast" is inspired by new modern techno-capitalism era where quantities are not calculated on needs but on abundance, and the speed of technology development force the human being to manage lot of different inputs simultaneously. The idea is to create an artificial environment similar to nature through the movement of sound, helping us to slow down and realizing that all animals live with nature timing.
The composition is built around a harmonic guitar chord progression that constantly grows at the end of every pattern. It goes from really soft to almost noise sound thanks to electronic filters that are triggered by guitar's dynamic touch and five matrix combinations. The piece aims at drawing the listener into an increasingly complex acoustic space where synthetic and acoustic sounds collide and merge to become one. The eternal struggle between man and nature can only end discovering that they are one and the same.

MATTERS 8

Daniel Mayer
Institute of Electronic Music and Acoustics /
University of Music and Performing Arts Graz

Materia (lat., substance, cause) – what other, than sound, could be the matter of music? Of course much more: rhythmic, harmonic and melodic structures, every kind of music, even rests. Possible, but I wouldn't like to premise that as given, whose novel malleability, owed to the computer, opens so many spaces as hardly anything else in the history of music. Moreover: no imaginations and ideas that detract from the essence.

Gérard Grisey: " … our model is sound not literature, sound not mathematics, sound not theatre, visual arts, quantum physics, geology, astrology or acupuncture."

Sound as mutual matter, it shall determine everything else: constellation and process, they ought to emerge from it, equitable, because without unfolding in time even the most sounding remains silent.

How do I find what I like?

Not at all, as I like what I find and I'm searching without knowing for what. It appears and queries me wordlessly, the talk develops within the experiment, the algorithms of transformation and organisation. Whatever in the end maybe – only just – can pass or, simpler then, can't pass in the face of that, what already exists and whereby the new scratches along trundling – that is determined by another matter: me – and in turn not; contingent and only seemingly private are memory and decision.

10:00–12:00 **Paper & Poster/Demo session 4**

Quantifying Pitch Drift in Unaccompanied Solo Singing: A Computational Examination through Density-Based Clustering

Sepideh Shafiei (Cu Test Inc)
Shapour Hakam (Cu Test Inc)
Arvin Nick (Cu Test Inc)

Comparing Audio Boundary Annotation of Vocal Polyphony: Experts, Non-experts, and Algorithms

Mirjam E. Visscher (Utrecht University)
Frans Wiering (Utrecht University)

Algorithmic Analysis and Harmonic Database Generation for Musicologist

Anaïs BINET (Université de Bordeaux / LaBRI)
Myriam Desainte-catherine (Université de Bordeaux / LaBRI)

Exploring Perde-Space: 3-D Tonnetz Visualization for Makam Music

Recep Gül (Istanbul Technical University)
Ozan Baysal (Istanbul Technical University)
Yavuz Buruk (Istanbul Technical University)
Yusuf Can Şeftali (Istanbul Technical University)

Exploring Immersion and Spatial Audio Perception in Concerts for Hearing-Impaired and Normal Hearing Audiences

Jesper Andersen (The Royal Danish Academy of Music)
Stefania Serafin (Aalborg)

The Elusive Nature of Inattentional Deafness: Assessing the Influence of Visual Attention on Background Music Perception in TV Programmes

Roser Batlle-Roca (Universitat Pompeu Fabra)
Diana Tyman (Universitat Pompeu Fabra)
Blai Meléndez-Catalán (BMAT Licensing S.L.)
Emilio Molina (BMAT)
Xavier Serra (Universitat Pompeu Fabra)
Perfecto Herrera-Boyer (Universitat Pompeu Fabra)

Statistical Analysis of Musical Features for Emotional Semantic Differentiation in Human and AI Databases

Francisco Braga (NOVA School of Science and Technology)
Jorge F Forero (Ludique)
Gilberto Bernardes (INESC TEC & University of Porto, Faculty of Engineering)

Temporal Analysis of Emotion Perception in Film Music: Insights from the FME-24 Dataset

Ruby O.N Crocker (Queen Mary University of London)
George Fazekas (QMUL)

BiSAID: Bipolar Semantic Adjectives Icons and Earcons Dataset

Zijing Cao (University of Porto)
Gilberto Bernardes (INESC TEC & University of Porto, Faculty of Engineering)
António Sá Pinto (INESC TEC & University of Porto, Faculty of Engineering)

"Back in my day...": a Preliminary Study on the Differences in Generational Groups Perception of Musically-evoked Emotion

Pedro L. Louro (DEI - FCTUC, University of Coimbra)
Renato Panda (Ci2.ipt, CISUC)

Exploring the Emergence of Beat Induction Using a 'Swarm of Onsets' Generative Model

Nolan Lem (Stanford University)

12:00 **Concert session 2**

wood, metal and tension - for violin and electronics

Cárthach Ó Nuanáin (Cork School of Music)

Cryoconite

Enrico Dorigatti (University of Portsmouth)

mirrored: ceilings, floors, walls

youngjae cho

TransVariations – Music beyond the limits of time and technology

Oeyvind Brandtsegg (Norwegian University of Science and Technology)
Alfonso Benetti (University of Aveiro)
Trond Engum (Norwegian University of Science and Technology)
Francisco Monteiro (Instituto Politécnico do Porto)

14:30 Keynote Speaker

Underwater Soundscapes: fostering connection and awareness through sonic immersion
Francisca Rocha Gonçalves

15:30 - 16:30 **Paper & Poster/Demo session 5**

Beamix: Application of an Ambisonics Filter Bank for the Creation of Directional Effects

Thibaut Carpentier (Ircam)

Toward a Novel Set of Pinna Anthropometric Features for Individualizing Head-Related Transfer Functions

Davide Fantini (Università degli Studi di Milano)
Stavros Ntalampiras (University of Milan)
Giorgio Presti (Università degli Studi di Milano)
Federico Avanzini (Università degli Studi di Milano)

A Speaker Agnostic Approach to Spatialisation in Electroacoustic Music

Stefano Catena (De Montfort University)
Henrik Frisk (Royal College of Music Stockholm)

Csound vs. ChucK: Sound Generation for XR Multi-Agent Audio Systems in the Meta Quest 3 Using the Unity Game Engine

Pedro P Lucas (University of Oslo)
Stefano Fasciani (University of Oslo)
Kyrre Glette (Oslo University)

Comparing Four 360-Degree Cameras for Spatial Video Recording and Analysis

Jinyue Guo (University of Oslo)
Maham Riaz (University of Oslo)
Alexander Refsum Jensenius (University of Oslo)

Exploring Convolutional Neural Networks for Multimodal Classification of Expressive Piano Performance

Anna-Maria Christodoulou (University of Oslo)
Sagar Dutta (University of Oslo)
Olivier Lartillot (University of Oslo)
Kyrre Glette (Oslo University)
Alexander Refsum Jensenius (University of Oslo)

Combining Hand Pose Estimation with Audio for Noise-robust Piano Score Following

Kaitlin Pet (Indiana University)
Akira Maezawa (Yamaha Corporation)

IEMI: an Immersive, Evolving Motion-controlled Installation

Ahmet Emin Memis (University of Oslo)
Tejaswinee Kelkar (University of Oslo)
A Cover Image Composition Method for Music Content Using Multimodal Image Retrieval and Cropping Techniques

Takayuki Nakatsuka (National Institute of Advanced Industrial Science and Technology (AIST))
Masahiro Hamasaki (National Institute of Advanced Industrial Science and Technology (AIST))
Masataka Goto (National Institute of Advanced Industrial Science and Technology (AIST))

16:30 ***Installation***

Transimmanency - Symbiotic Treatments, Trans-interactive installation with web and bright resonant objects

Lorenzo Ballerini (Conservatory of Trapani)
Alberto Gatti (Conservatorio di Musica Giacomo Puccini)
Giuseppe Ernandez (Conservatory of Trapani)
Massimo Reina (Conservatory of Trapani)

20:30 ***Conference Dinner***

KEY NOTE

UNDERWATER SOUNDSCAPES: FOSTERING CONNECTION AND AWARENESS THROUGH SONIC IMMERSION

FRANCISCA ROCHA GONÇALVES

DAY 2

UNDERWATER SOUNDSCAPES:

fostering connection and awareness through sonic immersion

FRANCISCA ROCHA GONÇALVES

In this keynote, I will explore the concept of immersion applied to my personal artistic practice, specifically through the medium of underwater sound recordings. My work connects with a growing awareness into the experience of being fully present in the moment—both during the process of capturing the subtle, often unnoticed sounds of the underwater world, and in the deep, attentive listening that follows.

Sound, as an artistic and sensory medium, has the unique ability to induce a state of intense focus, allowing us to transcend our immediate surroundings and connect with the environment on a more intimate level. The idea is to examine how such immersive practices can foster a heightened awareness and appreciation for the sonic world around us, nurturing a deeper connection with our environment, while promoting nature-connectedness.

By integrating this hyperfocus into our daily lives, we can elevate our presence and mindfulness, using our acute attention to sound as a means to deepen our environmental appreciation. I will draw from my artistic practice to share insights, including examples of how underwater soundscapes have been essential in creating this personal immersive experience. We will examine the impact of attentive listening, and the broader implications for using this "superpower" of deep tuning to enrich our daily lives and strengthen our connection to the natural world.

Exploring the intersection of art, sound, and attention, the objective is to discover how the practice of immersive listening can eventually transform our perception of the world and our place within it.

FRANCISCA ROCHA GONÇALVES
is a researcher from Porto, currently living in Berlin. Recently worked on the AQUATAG project at the IGB Leibniz Institute of Freshwater Ecology and Inland Fisheries. Has a background in biological sciences with a degree in Veterinary Medicine from ICBAS (University of Porto), a Multimedia Master in Interactive Music and Sound Design from FEUP (University of Porto), and a PhD in Digital Media from FEUP (University of Porto). The research focuses on acoustic ecology in artistic creation as a tool for environmental awareness concerning underwater soundscapes. Developing artistic artefacts that reveal the problem of noise pollution in underwater environments is possible to understand changes in vibration and particle motion, both vital components in aquatic life.
Combining interests in sound, technology, art and science, it aims to raise societal and environmental awareness through artistic practices and sound art. A great passion for biology and music led to a demand to find synergies between nature and sound. Bridging these two worlds, Francisca attempts to find new musical approaches, not only for musical compositions but also for live performances.
Presently is working for ICARUS, collaborating with Dr Johannes Goessling. ICARUS is an arts--science project that focuses on the photonic properties of diatoms and creates upscaled ice sculptures of their internal shells. Co-founder of the artistic collective Openfield Creative Lab. Co-founder of Ocean Soundscape Awareness project - ØSAW.

PAPER & POSTER /DEMO SESSION 4

/ Quantifying Pitch Drift in Unaccompanied Solo Singing: A Computational Examination through Density-Based Clustering

/ Comparing Audio Boundary Annotation of Vocal Polyphony: Experts, Non-experts, and Algorithms

/ Exploring Perde-Space: 3-D Tonnetz Visualization for Makam Music

/ Exploring Immersion and Spatial Audio Perception in Concerts for Hearing-Impaired and Normal Hearing Audiences

/ The Elusive Nature of Inattentional Deafness: Assessing the Influence of Visual Attention on Background Music Perception in TV Programmes

/ Statistical Analysis of Musical Features for Emotional Semantic Differentiation in Human and AI Databases

/ Temporal Analysis of Emotion Perception in Film Music: Insights from the FME-24 Dataset

/ BiSAID: Bipolar Semantic Adjectives Icons and Earcons Dataset

/ "Back in my day...": a Preliminary Study on the Differences in Generational Groups Perception of Musically-evoked Emotion

/ Exploring the Emergence of Beat Induction Using a 'Swarm of Onsets' Generative Model

QUANTIFYING PITCH DRIFT IN UNACCOMPANIED SOLO SINGING: A COMPUTATIONAL EXAMINATION THROUGH DENSITY-BASED CLUSTERING

Sepideh SHAFIEI[1], **Shapour HAKAM**[2], and **Arvin NICK**[3]

[1,2,3]**Cu Test Inc**, Berkeley, CA USA

ABSTRACT

This research delves into the nuanced phenomenon of pitch drift observed in unaccompanied vocal performances, wherein soloists inadvertently deviate from the initial pitch during protracted presentations. The manifestation of pitch drift depends on different factors, including the skill level of the performer, as well as the duration and intricacy of the musical rendition. In response to this, our study introduces a computational methodology designed to measure pitch drift and changes in intonation throughout unaccompanied vocal performances. Leveraging pitch histogram analysis in conjunction with DBSCAN clustering techniques, this innovative approach not only sheds light on the intricate dynamics of pitch variation but also provides a quantitative and visual framework for the assessment of such deviations in the realm of solo vocal expression. Through this comprehensive analysis, the study contributes to a deeper understanding of the subtle intricacies involved in pitch control and intonation changes during unaccompanied vocal presentations, which can be used as a visual tool for the vocalists for self-assessment. This novel approach also helps significantly in automatic transcription of vocal pieces with pitch drift.

1. INTRODUCTION

Intonation drift, a hallmark of unaccompanied singing, poses a significant challenge in Music Information Retrieval (MIR) tasks like automatic transcription. This phenomenon, characterized by gradual deviations from a reference pitch, usually in a downward direction, throughout a performance, has been explored in various contexts, including both choral and solo singing [1], [2], and [3]. While harmonic progression has been implicated in choral drift [4], and [5], the underlying mechanisms governing solo drift remain poorly understood. This paper presents a novel investigation into the computational measurement and characterization of intonation drift in solo singing. Leveraging a 5':32" vocal piece exhibiting pronounced drift, we develop a robust methodology for quantifying pitch deviations and analyzing their temporal and musical correlations. Our findings shed light on the nature of drift in solo singing and pave the way for improved automatic transcription and analysis of unaccompanied vocal music.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

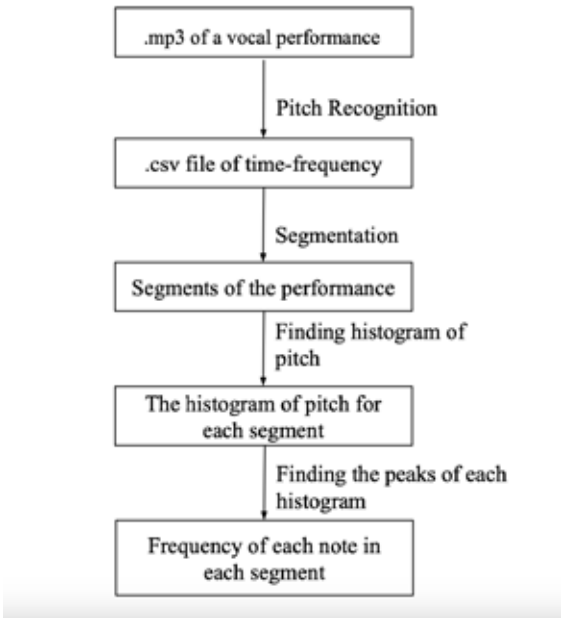


Figure 1. Steps in Data Preprocessing

The methodology is discussed in sections 2 to 4. In order to explain the detail we have used an example of a performance [1] with duration 5':32" (Example 1). It starts with an introduction for about 48 seconds, before the main part starts at the second 00:52". The whole piece has sixteen sentences, which are separated with long silences. The first two sentences are the introduction. A transcription of the whole performance can be found in the corresponding author's dissertation [6]: 69-71.

2. BACKGROUND AND PREPROCESSING

In this section, we present the building blocks of the data preprocessing together with all the needed musical and computational background. Figure 1 gives an overview of the preparation process in this section.

2.1 Pitch Recognition

There are different algorithms for pitch recognition. All these algorithms use the fundamental frequency of the

[1] The mp3 files for the performance in Example 1 is available in the github repository of the software along with some other examples from various musical cultures: https://github.com/SepiSha/PitchDrift

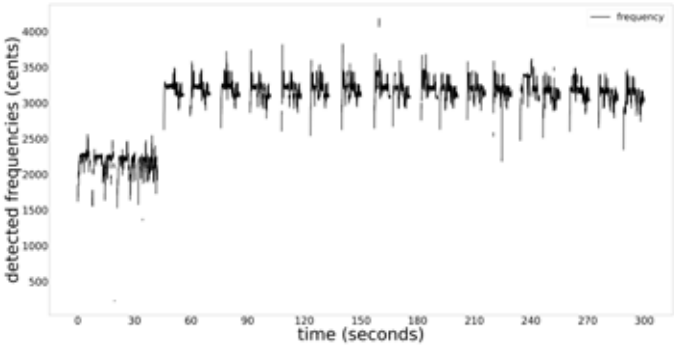


Figure 2. Time-Frequency for Example 1

sound to quantify pitch. Fundamental frequency is the lowest frequency of the sound wave and corresponds to its most dominant perceived pitch. Monophonic voice pitch estimation algorithms have been evaluated and discussed extensively in MIR literature. Gomez et al. have evaluated both algorithms CREPE [7] and pYIN [8] as state-of-the-art pitch recognition methods for monophonic voice [9]. They have evaluated the results of both of these algorithms on iKala dataset and obtained almost similar results for both pYIN and CREPE: 91% of Raw Pitch Accuracy for pYIN for monophonic voice and 90.5% accuracy for CREPE. Among other pitch recognition methods which can be used for monophonic sound one can mention SPICE [10], which is based on a self-supervised learning technique. We chose pYIN for this study, and we have done manual octave-error correction when necessary. Since we use the output of the pitch recognition method (pYIN) as an input for pitch recognition in *.csv* format, with minimum effort, one can replace the current pitch recognition method with any other algorithm for experiment or future improvement.

To make the pitch file we used Sonic Annotator for pitch recognition with the following parameters: step size of 256, block size of 2048, low amplitude suppression of 0.1, onset sensitivity of 0.7, prune threshold of 0.1, and threshold distribution of 2. Figure 2 shows the the time-frequency graph of Example 1. We have changed the frequency of the sound (Hz) to the cents system. Each octave is 1200 cents. The following formula gives the interval between the two notes with the frequencies f_1 and f_2, in cents.

$$cent = 1200 \times \log_2(\frac{f_2}{f_1}) \tag{1}$$

2.2 Identifying Musical Phrases

Phrasing and segmentation are crucial elements in music theory and cognitive musicology, helping to understand how listeners perceive and interpret musical structure. These concepts involve the ways in which music is divided into comprehensible units, influencing both the composition and perception of music. Segmentation is the process of dividing a musical piece into segments or sections based on criteria such as melody, harmony, rhythm, and texture such that it conveys a sense of beginning, continuation, and closure. These concepts are foundational for understanding musical form and expression, serving as the basis for more complex analyses of musical structure. There are various approaches to segmentation of music: The structuralist approach involves analyzing the syntactic rules governing musical structure. Leonard Meyer challenges the structuralist's premise of objective abstraction incorporating elements of musical education and cultural context that are both more subjective and more diverse [11]. He emphasized the importance of expectation in musical experience, suggesting that musical meaning arises from the interplay between learned patterns and their fulfillment or violation. This perspective highlights the importance of cultural and historical contexts in shaping musical understanding [12].

Cognitive approaches focus on how listeners mentally process and organize musical information. David Temperly uses computational approach to fundamental questions about music cognition [13]. Empirical studies often involve experiments where listeners' segmentations are recorded and analyzed. These studies provide insights into how listeners perceive and organize musical elements, revealing patterns that align with theoretical predictions and uncovering new dimensions of musical perception.

Computational models employ machine learning techniques to simulate human segmentation processes, offering a powerful tool for testing theoretical hypotheses and exploring new musical domains. Cross-cultural studies of phrasing and segmentation reveal both universal principles and culture-specific patterns. Genre-specific analyses highlight how different musical styles employ unique segmentation strategies, reflecting their distinct aesthetic and structural conventions.

Integrative theories such as Lerdahl and Jackendoff's Generative Theory of Tonal Music (GTTM) provides a formal framework for analyzing the hierarchical structure of tonal music, combining insights from music theory, cognitive science, and linguistics. The GTTM's main components—Grouping Structure, Metrical Structure, Time-Span Reduction, and Prolongational Reduction which in this framework signifies hierarchical pitch connection—offer a comprehensive model for understanding how listeners parse musical surfaces into structured representations [14].

Since we started this research as part of a larger project on the computational approach to Iranian classical vocal music, we have noted that the phrasing in this genre often follows the structure of classical poetry, with phrases frequently separated by long silences. In this paper, we perform segmentation based on these relatively long silences between the sentences sung by the vocalist. This method helps us identify musical phrases, or sentences. For future reference in this paper, we will call this type of segmentation *Method I*. An alternative approach for computing pitch drift is to consider congruent segments, i.e., every n seconds. We refer to this equal-duration partition of the vocal piece as *Method II*. In this example, we have chosen the length of each segment to be $T/24$, where T is the total duration of the piece in Example 1. In the next section, we consider both methods. In the first case, the segments vary

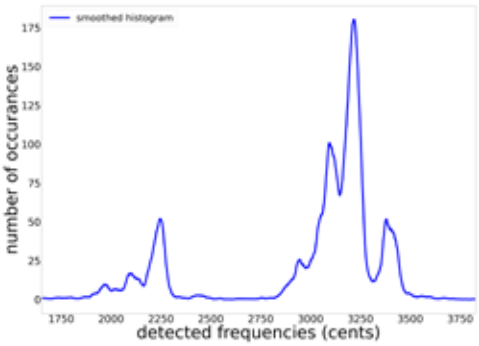


Figure 3. Pitch Histogram for Example 1

in length depending on the duration of the sentences, while in the second method, the segments have equal duration. There is always the potential to employ various types of segmentation depending on the genre of music and differing theoretical approaches, which can be explored in future research.

2.3 Pitch Histogram

The next step in preparation of the data points is to find the histograms of the performed pitches for each segment. In order to find the histogram, we find the total number of occurrences of each fundamental frequency. Figure 3 shows the pitch histogram of Example 1. The vertical axis shows the proportional total duration of each frequency (in cents). Pitch histogram has been used extensively in MIR literature. For example, Bozkurt et al. have used it for analysis of Turkish Makam Music [15]. In the musical traditions where intervals and note frequencies are not absolute, using pitch histograms are useful in identification of the frequency of each note and the intervals between the notes in the performed pieces. The use of pitch histogram in finding intervals of a performed vocal piece in Iranian music has been discussed extensively in [16] and [17]. Koduri et al. have used pitch histograms in analysis of intonation for Carnatic music [18].

2.4 Gaussian Model of the Peaks

We smooth the histogram in order to find the exact peak of each mountain by finding the moving average to smooth out short-term fluctuations and highlight the general trend of the data and we get a semi-Gaussian curve. Choosing reasonable peaks is very crucial. To find the peak of each mountain, which represents the median of the performed pitch for each "note", we need to find the range of the mountain which is itself a challenging task and is discussed in [16]. After finding the range of each mountain, we model each mountain by a tilted Gaussian curve so that we can find a better peak. We fit the following curve to our data to find the parameters $c_1, \ldots, c_5$

$$y = c_1 + c_2 x + c_3 e^{-(x-c_4)^2/c_5} \tag{2}$$

After finding the peak for each mountain in the histogram

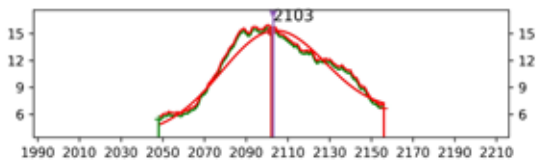


Figure 4. Gaussian peak fit for the first significant peak of the pitch histogram Example 1

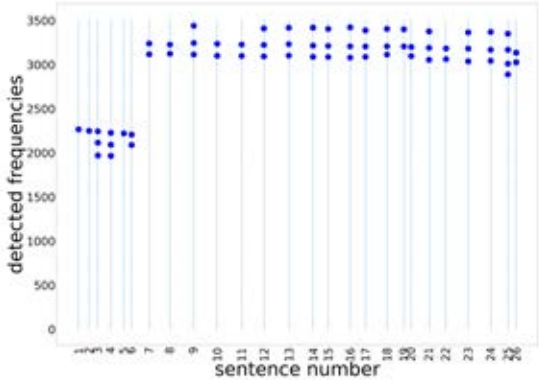


Figure 5. The main detected frequencies for each sentence in Example 1 using *Method I*

of audio, we have the frequency of every note in each sentence in cents. Since we only want reliable estimates, we have picked the mountains that have at least certain heights, so that we have enough data for fitting a Gaussian curve. Figure 4 shows the result of our Gaussian peak fit for the first significant peak of the pitch histogram in Example 1.

3. EXPLANATION OF THE DATA

After performing the preprocessing which was explained in sections 2.1 through 2.4, we have gathered the frequencies associated to the main notes of each segment. Figure 5 shows the data points gathered using *Method I* for segmentation. The horizontal axis shows the segment number and the vertical axis shows the frequency of the main notes in each sentence in cents. Since in *Method I* the duration of the sentences varies, the numbers on the x-axis are not equidistant, the distance of each number from its consecutive number is proportional to its duration. If we use *Method II* and find the frequency of the notes in the same size intervals (every n seconds), we will get Figure 6 for the vocal performance in Example 1.

4. CLUSTERING THE PEAKS USING DBSACN

To see the pattern of drift in various notes throughout the piece, we first sort the frequency of the notes for each segment of the piece. Assume that we have n segments $\{s_1, ..., s_n\}$. For each of these segments $s_i, 1 < i < n$, we will have a number of frequency peaks i_k: $p_{i_1}, ..., p_{i_k}$, the number of the peaks varies from one segment to the other.

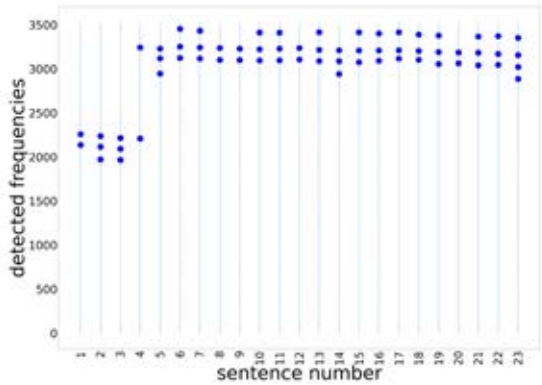


Figure 6. The main detected frequencies for each sentence in Example 1 using *Method II*

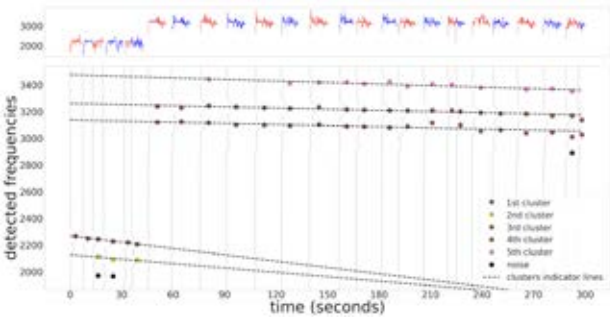


Figure 7. Clustering of the frequencies using DBSCAN, and modeling each cluster with linear regression, using *Method I* for segmentation

For example, in Figure 6, $i_1 = 2$ and $i_2 = 3$, which means that we have two peaks in the first segment, and three peaks in the second segment. We need to cluster these frequency points p_{i_k} for $\forall i, k$, so that we have all the fundamental frequencies associated to a specific note throughout the performance in one class.

In order to cluster the frequency data points we used Density-based spatial clustering (DBSCAN) from *scikit* in Python. We then use linear regression to model each cluster. Figures 7 and 8 show the result of this system for Example 1 using Method I and Method II for segmentation respectively. As can be seen in these Figures, we have detected 5 cluster of points, each corresponds to a note. Each cluster is marked in the graph with a separate color. The first two clusters associated to the lower frequency notes only appear at the beginning of the piece and have fewer samples comparing to the other three clusters. The slope of the lines in the linear regression shows the pattern of change in the tuning during the course of the performance. This method for quantification of pitch drift can be very helpful in automatic transcription of unaccompanied vocal pieces, especially when the pitch drift is significant. As can be seen from the two Figures 7 and 8 the method of segmentation did not effect the result significantly.

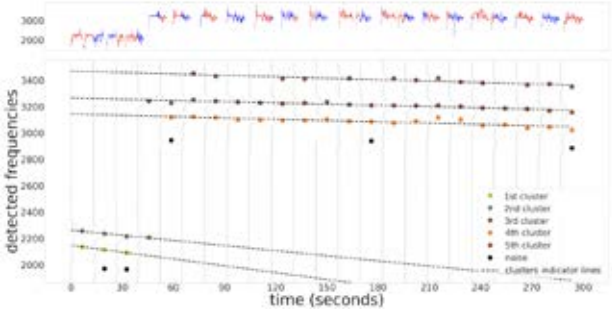


Figure 8. Clustering of the frequencies using DBSCAN, and modeling each cluster with linear regression, using *Method II* for segmentation

Acknowledgments

This work was started as part of the first author's PhD dissertation under the supervision of Prof. Blum in the Music Department of the City University of New York.

5. REFERENCES

[1] M. Mauch, K. Frieler, and S. Dixon, . "Intonation in unaccompanied singing: Accuracy, drift, and a model of reference pitch memory." *The Journal of the Acoustical Society of America*, 136(1), pp.401-411. 2014.

[2] J. Dai, M. Mauch, and S Dixon. "Analysis of intonation trajectories in solo singing." In *Proceedings of the 16th ISMIR Conference* (421: 29). October 2015.

[3] J. Dai, and S. Dixon, 2019. "Intonation trajectories within tones in unaccompanied soprano, alto, tenor, bass quartet singing." *The Journal of the Acoustical Society of America*, 146(2), 1005-1014.

[4] H. Terasawa. "Pitch drift in choral music." Music 221A final paper, Center for Computer Research in Music and Acoustics, at Stanford University, CA. Available at https://ccrma. stanford.edu/hiroko/pitchdrift/paper221A. pdf (Last viewed 5 June 2014).

[5] D. Howard.. "Intonation drift in a capella soprano, alto, tenor, bass quartet singing with key modulation." *Journal of Voice*, 21(3), pp.300-315. 2007.

[6] S. Shafiei, *Extracting Theory from Practice: A Computational Analysis of the Persian Radif*. PhD dissertation at the Graduate Center, City University of New York. 2021.

[7] J. Kim, J. Salamon, P. Li, and J. Pablo Bello. "CREPE: A Convolutional Representation for Pitch Estimation." In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*: 161-165. IEEE. 2018 .

[8] M. Mauch, and S. Dixon. "pYIN: A fundamental frequency estimator using probabilistic threshold distributions." In *2014 IEEE International Conference on*

Acoustics, Speech and Signal Processing (ICASSP), pp. 659-663. IEEE, 2014.

[9] E. Gómez, M. Blaauw, J. Bonada, P. Chandna, and H. Cuesta. “Deep Learning for Singing Processing: Achievements, Challenges and Impact on Singers and Listeners." arXiv preprint. arXiv:1807.03046. 2018.

[10] Gfeller, Beat, et al. "SPICE: Self-supervised pitch estimation." *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 28 (2020): 1118-1128.

[11] Swain, Joseph P. "Leonard Meyer’s New Theory of Style." *Music Analysis* 11 (1992): 335-354.

[12] Meyer, L. B. 1956. and Meaning in Music. University of Chicago Press

[13] Temperley, D., 2004. *The cognition of basic musical structures*. MIT press.

[14] Lerdahl, F., and Jackendoff, R. 1983. *A Generative Theory of Tonal Music*. MIT Press.

[15] B. Bozkurt. “Pitch Histogram Based Analysis of Makam Music in Turkey.” Paper presented at *Les Corpus de l’oralité Colloquium, Strasbourg*. 2011.

[16] S. Shafiei. “An analysis of Iranian Music Intervals based on Pitch Histogram. arXiv preprint arXiv:2108.01283.

[17] S. Shafiei, "Analysis of Vocal Ornamentation in Iranian Classical Music." In *Sound Music Conference Proceedings*. 2019.

[18] Koduri, Gopala Krishna, Joan Serrà Julià, and Xavier Serra. "Characterization of intonation in carnatic music by parametrizing pitch histograms." Gouyon F, Herrera P, Martins LG, Müller M. ISMIR 2012: Proceedings of the 13th International Society for Music Information Retrieval Conference; 2012 Oct 8-12; Porto, Portugal. Porto: FEUP Ediçoes, 2012.. International Society for Music Information Retrieval (ISMIR), 2012.

COMPARING AUDIO BOUNDARY ANNOTATION OF VOCAL POLYPHONY: EXPERTS, NON-EXPERTS, AND ALGORITHMS

Mirjam VISSCHER (m.e.visscher@uu.nl) [1] and **Frans WIERING** (f.wiering@uu.nl) [1]

[1] *Department of Information and Computing Sciences*, **Utrecht University**, Netherlands

ABSTRACT

It is a challenging computational problem to perform segmentation on vocal polyphony from the Renaissance and early Baroque. In this genre, boundaries between segments are often hidden by overlapping voices. To test algorithms for segmentation, we need boundary annotations by humans as a ground truth, but experts in this field are rare and short on time.

Our study aims to evaluate the effectiveness of segmentation algorithms on vocal polyphony using both expert and non-expert annotations. For this, we collect boundary annotations by human experts and non-experts on polyphony. Then, we compare the annotations by the two groups to see whether we can use segmentations by non-experts instead of experts. Finally, we use the expert annotations to evaluate different segmentation algorithms from the MSAF library by Nieto and Bello.

The results show that the performance of non-experts comes quite close to that of experts, whereas the tested algorithms are not yet able to perform the task at a similar level. We conclude that non-expert annotations are adequate to act as ground truth for evaluating boundary detectors on vocal polyphony and we present next steps to create a larger dataset for such evaluations.

1. INTRODUCTION

The rich amount of computational methods developed in more than 25 years of MIR research presents a fascinating opportunity for music research. Can these methods be used to gain a deeper understanding of musical structure, music history, and musical cultures around the world? This raises the question of how generic those methods are, since the majority have been developed for and tested on contemporary popular music. How does this affect their performance or even validity when applied to different musics?

One emerging line of computational music research focuses on historical processes in Western music [1–3]. Within this broad area, our work concentrates on late Renaissance and early Baroque polyphony. Polyphony proves to be particularly challenging due to the relative homogeneity of the textures and the overlapping of voices. Also, during this period, music underwent many changes along a number of dimensions, one of which is tonal structure. Many musicologists claim that the modal system (based on melodic relationships) was replaced by modern harmonic tonality (based on chord relationships) during the 16th century. Others have different views [4]. Such conflicting accounts are based on the detailed analysis of music theorists and selected key compositions. It would be interesting to investigate if such a close reading of individual items could be complemented by a distant listening approach [5], and if this would shed a new light on the historical developments of the time.

When conducting research on tonal structure, the most logical approach would seem to be to analyse symbolic encodings. But since these are rather scarce (see e.g. figure 1 in [3]) and moreover employ a variety of encoding formats, it makes sense to turn to audio recordings instead, of which there are a much higher number thanks to the wide interest in early music from the 1960s onward. It is the aim of our CANTOSTREAM project [1] to investigate to what extent meaningful historical patterns can be found in audio recordings with the help of MIR methods.

A first step towards this this long-term aim is to divide each composition automatically into sections at meaningful boundaries. But what are those meaningful boundaries? This research presents a method to collect perceptual data from expert and non-expert listeners, delivering a dataset of boundary annotations, and evaluates six boundary detectors to gain an understanding of their potential for the task. Specifically, the research questions are:

1. Can we use non-expert annotations of boundaries to improve segmentation algorithms for vocal polyphony?
2. How does a selection of current boundary algorithms perform on vocal vocal polyphony?

We will show that it is possible to collect boundary annotations for retrieval experiments, employing a minimalist and economic setup.

2. BACKGROUND

2.1 Boundaries in Vocal Polyphony

Boundaries are points in music that provide a sense of closure to a section of a composition. In vocal polyphony, it is important to make a distinction between boundaries as a section closure on the one hand, and cadences as described

[1] https://www.uu.nl/en/research/interaction/music-information-computing/projects

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Algorithm	Full name	Main characteristics
Foote [6]	Foote	Novelty
CNMF [7]	Convex non-negative matrix factorization	Timbre, clustering decomposition matrices
OLDA [8]	Ordinal linear discriminant analysis	Chord, timbre, pitch, timing
Scluster [9]	Laplacian segmentation, spectral clustering	Local timbre and long-term repetition
SF [10]	Structural features	Harmonic pitch class profiles; novelty, homogeneity, repetition
VMO [11]	Variable Markov Oracle	Harmony, timbre

Table 1. MSAF algorithms, as named in the library

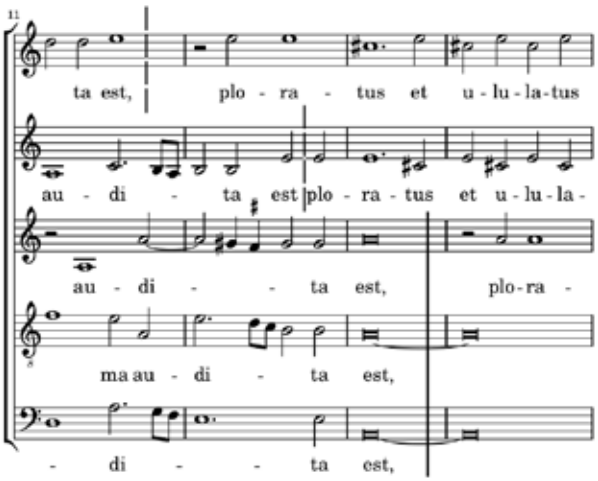


Figure 1. A polyphonic cadence in Giaches de Wert, Vox in Rama, mm 11-14. The vertical line in bar 13 indicates the cadence, the dotted lines indicate the individual endings.

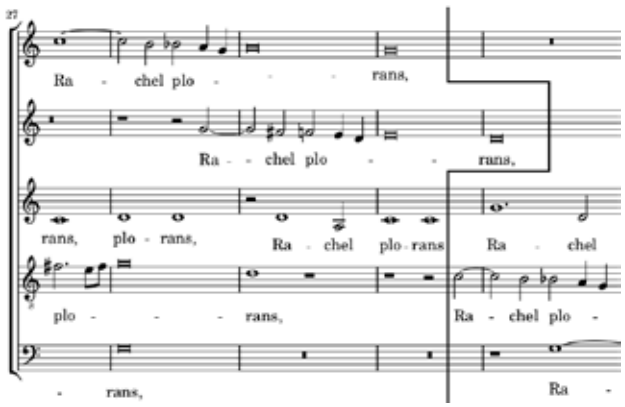


Figure 2. A boundary without cadential patterns in Giaches de Wert, Vox in Rama, mm 27-31.

in music theoretical sources on the other [12–14]. Boundaries can be formed by using a number of compositional techniques, such as change of texture, change in the combination of voices, change of meter, rests and cadences. Figure 1 provides an example of a typical polyphonic cadence in a motet by Giaches de Wert. The three lowest voices form a cadence on A in measure 13, while the two upper voices already have introduced the next phrase. This boundary is easy to recognise by ear, but most algorithms studied in this paper miss this boundary. An important part of the composer's art is to play with cadential structures to manipulate the listeners' expectations of closure. Not all cadence-like patterns are therefore strong boundaries, or even boundaries at all. Conversely, De Wert provides us with an example of a boundary without cadential elements in figure 2. Yet the majority of the participants in our experiment perceive a boundary between the first *Rachel plorans* and its repetition, as indicated by the solid line.

2.2 Automated Boundary Detection in Musical Audio

Boundary detection is usually regarded as the initial stage in audio-based music structure analysis, followed by structural grouping [15]. Three primary approaches were described by [16], reviewing the state of the art up to 2010, repetition-based, novelty-based, and homogeneity-based, to which a recent overview [17] has added regularity-based approaches. Additionally, [17] presents three main challenges in the field, namely subjectivity, ambiguity and hierarchy, affecting the annotator, the annotations and the musical structure respectively. Of these, subjectivity is the most important for our paper.

Both [16] and [17] present a comprehensive overview of segmentation methods. In our research, we focus on the subset of these methods provided by [15] which are implemented in their open-source music structure analysis framework (MSAF). [2] Table 1 displays the six boundary detection algorithms from this library, with their main characteristics.

2.3 Annotation of Musical Boundaries

There is a long history of experimentation with boundary perception in music [18–23]. From these experiments we learn a number of things. All authors recommend the use of surveys to gauge the expertise of their participants, for example the recent Goldsmiths Musical Sophistication Index [24]. However, this survey does not ask about experience with music from a specific period or genre.

Most authors used some form of tool support in their experiments. Recently, Bedoya [23] proposed the CosmoNote platform for the study of musical prosody, one aspect of which is segmentation (scored on a scale of 1-4). This platform is directed at citizen scientists creating their annotations online. They are supported by various visualisations and can review and correct their annotations.

[2] https://github.com/urinieto/msaf/

Composer	Work	Muziekweb.nl id
Luca Marenzio	Zefiro Torna (1584)	DBX12531-9
Orlande de Lassus	Pater Noster (1573)	DBX10350-13
Giovanni Pierluigi da Palestrina	Vergine tale è terra (1581)	DBX0438-37
Orlande de Lassus	Pauper sum ego (1573)	DBX0961-4
Orlande de Lassus	Justorum animae (1582)	DBX8123-20
Giovanni Pierluigi da Palestrina	Osculetur me osculo oris sui (1584)	DBX0438-1
Claudio Monteverdi	Or che 'l ciel e la terra (1638)	DBX12134-2
Giaches de Wert	Vox in Rama (1581)	DBX12064-4

Table 2. Recordings used for the experiment.

The importance of clear instructions for the participants is emphasised in [21]. It is important to hide visual cues that can influence the placement of boundaries [20, 21].

In such experiments, participants do not indicate a boundary at the exact same time although they respond to the same stimulus. Therefore, all authors use a window of synchrony [25]: "An interval of time over which response events are counted as occurring together", henceforth called window. Most boundary experiments set their window at +/- 3 seconds [15, 23], while [25] sets it at 2 seconds for the task of annotating arousal and valence.

3. METHOD

In our experiment, we collect boundary annotations by letting participants tap. We use these annotations to evaluate selected algorithms. The workflow of the experiment, as visualised in Figure 3, starts with the selection of compositions, continues with collecting annotations, and applying boundary detectors. Then, the workflow divides into two strands. The left strand is a cluster analysis to answer the question whether we can use non-expert annotations of boundaries to improve segmentation algorithms for vocal polyphony. The right strand evaluates the performance of non-experts and the MSAF algorithms respectively.

3.1 Selection of Musical Works

We select eight recorded performances of vocal polyphonic works composed around 1600. Table 2 lists the works, with a total duration of 36 minutes. We chose vocal music to keep timbre similar. There is a variety in polyphonic versus homophonic writing, in ambiguity in the boundaries, in the estimated difficulty to annotate the boundaries, in the tempo, and in the amount of voices. We select the performances for their restraint and professional level of the performers.

3.2 Preliminary Experiments

On three important elements of the experiment, the literature provides no conclusive recommendations. Therefore, we conduct preliminary experiments. First, we test boundary annotation on a printed score. This task took the expert participants about 30 minutes for 5 minutes of music. Some of the boundaries indicated seemed non-intuitive when compared to the perception of the performance, due to a music theoretical approach suggested by the presence of a score.

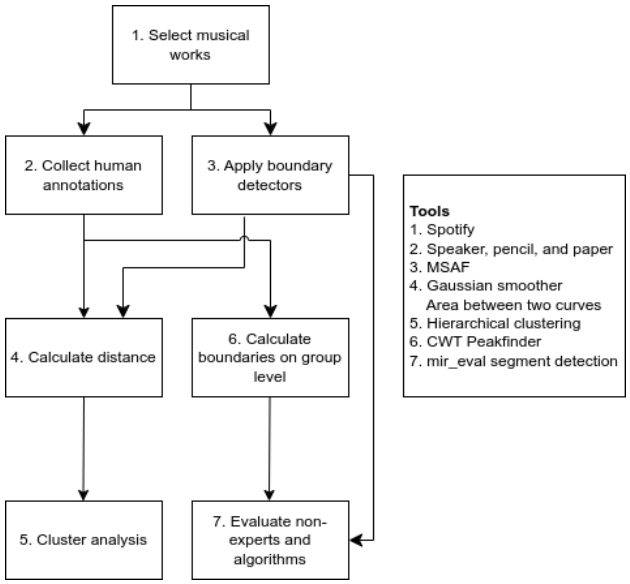


Figure 3. Workflow and tools used for this paper. The order corresponds to the order of the method.

As an alternative, we investigate a combination of audio and touch. Specifically, we test which type of touch would be most intuitive for a listener to use as an indicator of boundary strength in real-time: duration, pressure, amount of fingers pressing, or release of the touch when there is a boundary. Duration was perceived as the most intuitive method.

Finally, we test the process with participants having different levels of musical experience, ranging from no experience at all to several decades of professional experience. The boundaries indicated by all participants suggest a general agreement on the location of the boundaries, with varying preferences for granularity.

3.3 Collection of Annotations

For the main experiment, we ask the participants about their years of musical training, practice, and experience with Renaissance music using a custom survey. [3]

[3] The survey, a summary of the answers, the text used for the instructions, the anonymised annotation data

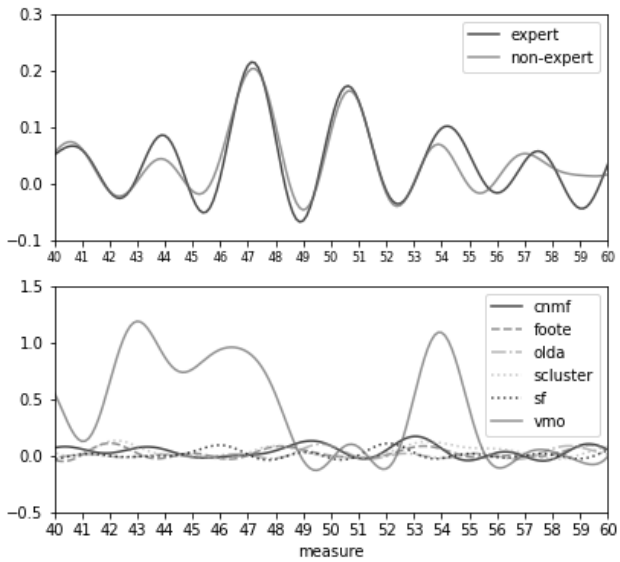


Figure 4. Gaussian curves of boundaries by participants and algorithms for Palestrina, *Vergine tale è terra* mm 40-60.

The participants listen to the music without looking at the score, and they tap the hand of the first author if they perceive a boundary, longer taps for strong boundaries, shorter taps for weak boundaries using a scale ranging from 1 to 4, and a resolution of one quarter note. The researcher records the boundaries and their weights on the score.

Prior to the experiment, we calibrate the strength of the boundaries: the first author asks the participant to tap the weights of 1, 2, 3, and 4 subsequently and provides feedback if the lengths are not clear. We use the first work to test and potentially correct the granularity preferences of the participant. We observed that each participant kept at roughly the same level of granularity and strategy during the session.

We decide not to give the participants the opportunity to re-listen and correct their annotations. First, because we are interested in a immediate, perceptual experience, whereas correction of annotation involves a analytical mode of thinking. Secondly, to keep the duration of the experiment within acceptable limits while still being able to annotate multiple compositions.

One annotation session is observed by the second author, with a focus on validity risks in music information retrieval [26]. The main observations are that the annotator follows the tapping of the participant accurately and without hesitation, and does not appear to steer the participant in a particular direction, whether consciously or unconsciously.

The second author provides reference annotations from a music analytical perspective, based on the score. He focuses on high- and mid-level structures, taking into account compositional techniques as well as the setting of the texts.

and the code used for the analysis are available on https://github.com/MirjamVisscher/cantostream_boundaries.

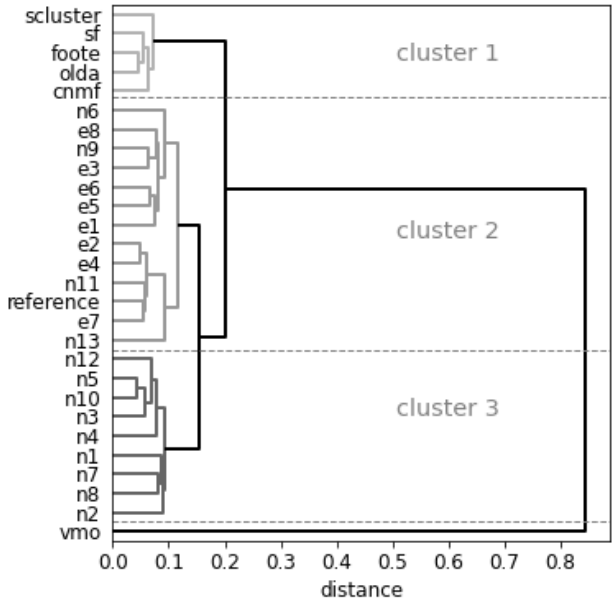


Figure 5. Clustering of distances between Gaussian curves of individual participants non-experts n1 to n13, experts e1 to e8, reference and algorithms.

3.4 Boundary Detectors

We apply the MSAF boundary detectors [15] as provided in the library with standard settings, the input format being wav files. While the algorithms do not give us a weight per boundary, they often provide more than one boundary per quarter note. We do not want to discard this information from the analysis and count number boundaries per quarter note and use this as a weight.

3.5 Analysing Distance between Participants and Algorithms

The participant annotations have a high resolution of one quarter note, and annotations based on the same musical cue may be several quarter notes apart. Therefore it does not make much sense to directly compare annotations. Instead, we smooth the time series using a Gaussian kernel with sigma = 0.002. Subsequently, we compute the pairwise distances between the Gaussian curves of all participants and algorithms [27]. Utilizing the resulting distance matrix, we analyze how individual participants and algorithms cluster through hierarchical cluster analysis using the Ward method from the SciPy implementation [28].

3.6 Compute Boundaries on Group Level

The second strand of the workflow involves identifying boundaries on the level of participant groups. For both experts and non-experts, we begin by averaging annotation weights. Then, we utilize a peakfinder with continuous wavelet transformation [29], applying a window of 4 quarter notes to correct for the spread in timing by participants around perceived boundaries.

	window = 8			window = 4			window = 2		
	P	R	F	P	R	F	P	R	F
reference	.92	.71	.80	.88	.68	.76	.80	.62	.70
non-experts	.81	.90	.85	.75	.83	.79	.65	.72	.68
Foote	.42	.39	.41	.23	.22	.23	.17	.16	.17
SF	.40	.34	.37	.19	.16	.17	.14	.12	.13
CNMF	.39	.52	.45	.21	.27	.24	.14	.18	.16
OLDA	.34	.34	.34	.25	.25	.25	.18	.18	.18
Scluster	.31	.40	.35	.17	.22	.19	.11	.14	.12
VMO	.06	.84	.10	.05	.73	.09	.04	.66	.08

Table 3. Evaluation of reference, non-experts, and MSAF algorithms for all Early8 works combined, evaluated against the expert ground truth.

3.7 Evaluate Non-Experts and Algorithms

We consider the boundaries by the expert participants to be the ground truth. We evaluate the group level boundaries by non-experts and the boundaries by the separate algorithms against this ground truth, using precision, recall and F1 score. Next, we compare the algorithm boundaries to the peaks of the experts. For the evaluation, we use the segment detection method [30] from mir_eval [4] with a window of synchrony of 8 quarter notes, comparable to the aforementioned window of 3 seconds. This means that a boundary is considered in agreement when it is within a window of 8 quarter notes from the expert peaks. From a musical standpoint a window of 8 quarter notes or 3 seconds is quite lenient. Therefore, we also evaluate with window sizes of 2 and 4 quarter notes to see how this impacts the outcomes. For the evaluation of the non-experts, we take into account both precision and recall. For the algorithms, we only take into account the precision since we can employ more than one algorithm, each one specialised in its own type of boundaries.

4. RESULTS

4.1 Collection of Annotations

We recruited 22 participants: 8 experts, 13 non-experts, and one reference participant with a music analysis background. On average, the experts have 30 years of musical training and 27 years of Renaissance practice, the non-experts have on average 3 years of musical training and no Renaissance practice. The experiment took place between September 2022 and January 2023. Eleven non-experts and all experts finished the complete experiment. Each session had a duration of 45 to 90 minutes, depending on the concentration and the desired breaks of the participants.

There were no noticeable changes in granularity after calibration of the granularity in the first work. Therefore, we decided to keep the results of the first work in the dataset.

As an example, figure 4 shows the Gaussian curves for Palestrina, *Vergine tale è terra*, the upper pane for the participants, and the lower pane for the algorithms. Experts and non-experts show a clear agreement on where the boundaries are, whereas the algorithms show no agreement at al. A similar picture emerges for all compositions.

4.2 Clustering of Participants and Algorithms

The clustered individual participants and algorithms in Figure 5 show a clear division between humans and computer. Cluster 1 contains all algorithms, except for VMO, which has a large distance to the other algorithms. Cluster 2 contains the reference, all experts and four non-experts; cluster 3 contains non-experts only. Due to its high density of boundaries, VMO is an outlier, as is shown in Figure 4.

4.3 Evaluation of the Humans and the Algorithms

We evaluate the boundaries of the non-experts, the reference, and the algorithms against the expert ground truth, using precision, recall and F1 score, and with 3 window sizes: 8 quarter notes, 4, and 2.

In Table 3, window = 8, we see that the non-experts have a precision of .81 and a recall of 0.90. This suggests that the non-experts generally find the same boundaries as the experts, although on a slightly more granular level. This is confirmed by a visual inspection of the annotations. The reference in turn has a precision of .92 and a somewhat lower recall of .71. This suggests that the reference yields similar boundaries as the experts, but on a higher structural level.

All algorithms have a much lower performance than the non-experts. The best performing algorithms are Foote, SF and CNMF, having a precision around .40.

We investigate the impact of window size on the performance of non-experts and algorithms by rerunning the evaluation with window sizes of 4 and 2. The non-experts show a moderate decrease, whereas the algorithms drop by a much larger degree. This suggests that, even if algorithms are able to find a boundary, they tend to place it on a different location than humans.

5. DISCUSSION

5.1 Not Every Boundary is a Cadence

As discussed in section 2.1, not every boundary in vocal polyphony is marked by a formal cadence: there are other

[4] https://github.com/craffel/mir_evaluators

Work	Cadences	Plagal	Non-cadences	Total
1	14	2	2	18
2	13	3	2	18
3	15	2	0	17
4	6	1	2	9
5	5	3	1	9
6	8	1	1	10
7	17	3	4	24
8	9	3	3	15
Total	87	18	15	120

Table 4. Expert boundaries classified by the second author into cadence and non-cadence types.

types of boundaries, too. In Table 4 we count the different types of boundaries that occur in the expert annotations (as assessed by the second author). The majority of boundaries is indeed marked by a cadence that follows period theory to a larger or smaller extent. Fifteen percent of the boundaries are plagal cadences (which are not described in the theoretical sources), and another 13 percent of the boundaries is not marked by a cadence at all. This finding has an important implication: if we were to segment compositions based on cadences alone, we would miss out on a significant part of the boundaries that expert listeners deem to be relevant.

5.2 Segmenting Homophonic Compositions

In complex vocal polyphony, cadences and other structural markers are overlapped by voices that follow their own course uninterrupted. This may be one important reason for the low performance of the MSAF algorithms. To put this in perspective, we tested the performance of the algorithms on two works of a more homophonic nature, *Innsbruck, ich muss dich lassen* by Heinrich Isaac, and *Ave verum corpus* by William Byrd. The reference annotations are provided by the first author, by means of a structural analysis as described in section 4.1. The evaluation results are shown in Table 5.

All algorithms in this additional experiment, except for VMO, show a precision between 0.77 and 0.94, and a recall between 0.43 and 0.69. As explained in section 3.8, the high precision scores are particularly interesting since this would potentially allow the combination of multiple detectors combined to increase recall. The outcomes also show that these detectors can be generalised beyond the repertoire they were trained on, namely popular and some classical music. Even though the timbral and tonal features of the homophonic compositions differ substantially from the training data, the algorithms appear able to deal with these. The most important challenge for improving the algorithms for vocal polyphony therefore seems to be the presence of overlapping voices in more complex compositions.

	P	R	F
SF	.94	.43	.59
OLDA	.87	.57	.69
Scluster	.86	.69	.76
Foote	.83	.54	.66
CNMF	.77	.69	.73
VMO	.06	1.00	.12

Table 5. Evaluation the MSAF algorithms for Isaac and Byrd using a window size of 8.

5.3 Experimental Design

Our experiment is primarily designed to efficiently capture perceived musical boundaries. To promote perception, we provide no score or other visual cues that can prompt analytical instead of perceptual annotations. We found that revision of the annotations by participants is not needed to obtain consistent results.

The choice for the task of tapping to music was deliberate, as it falls within the range of spontaneous gestures induced by music, whereas traditional computer interfaces can pose a barrier between perception and task execution. However, the quality of the data depends on the skills of the experimenter, and the setup of this experiment imposes limitations on the number of annotations possible.

Fortunately, [22] offers a model of how to scale up and collect consistent annotations in an online setting. For our purposes, it would be crucial to disable annotation review functionalities and not to use any score visualisation. Also we would recommend a simplified input mode that allows the annotation of a boundary through a single gesture, for example by tapping a touchpad and recording the duration of the tap.

5.4 Limitations

The number of participants in this experiment is limited to 8 experts and 13 non-experts, which invites further investigation on the optimum number of annotators. In addition, annotation of a wider range of works is important for gaining a better insight in early music segmentation. The eight selected musical works are all vocal: we have no information on how instrumental works would impact the results. The works are from a limited time span in the late Renaissance and early Baroque, we do not know how participants and algorithms would perform on works selected from a wider time span.

During the the experiment, we initially notated strong boundaries that were perceived on long notes, at the end of that specific note. Later, based on our first experiences, we decided to annotate boundaries on the exact location where the participant tapped. As a consequence, the strong boundaries annotated by the non-experts may appear simultaneously, while in reality they might have been tapped at slightly different moments.

For the MSAF algorithms, we have used the standard, out-of-the-box settings, using a fixed seed for stochastic

elements in the algorithms, to ensure that the results are reproducible. The algorithms might perform better with custom settings or after being retrained on a more fitting corpus.

6. CONCLUSION

This paper presents a new annotation dataset and answers two main questions. First, it examines whether non-expert annotations of boundaries are fit to improve segmentation algorithms for vocal polyphony. And secondly, it evaluates the performance of a selection of current boundary algorithms on vocal polyphony.

6.1 Agreement Between Non-Experts and Experts

When clustering, we expected three clearly separated clusters of experts versus non-experts versus algorithms. The participants do indeed form two clusters, but interestingly in one, experts and non-experts are mixed, while the other consists of non-experts only. This suggests that differences between experts and non-experts are not particularly strong.

In the evaluation, we treated experts and non-experts as two separate groups, and considered the expert annotations as the ground truth. We tested the precision and recall of the non-expert boundaries measured against the expert boundaries. The non-experts have a precision of 0.81 and a recall of 0.90. This is a noticeable but not a very large difference.

We conclude that non-expert annotations are close to expert annotations but not fully interchangeable. These findings are in line with [31], where experimental outcomes suggest that musically untrained listeners have the ability to process musical structures in a similar way as musical experts. Philips et al. [32] found supporting results, after comparing expert and non-expert segmentation in contemporary music, concluding that the performance of both groups was very similar.

6.2 Algorithms Boundaries are Far Removed from those by Experts

In our analysis, VMO forms its own cluster due to an unusual high amount of boundaries indicated. The other algorithms form another cluster, clearly separated from the human annotations.

Since the performance of non-experts is much higher than the performance of algorithms and fairly close to the expert ground truth, we conclude that non-expert annotations of boundaries in vocal polyphony are suitable to evaluate the performance of boundary detectors until one or more detectors reach the level of humans.

7. FUTURE WORK

Now we have established that non-expert annotations of boundaries are similar enough to expert annotations to be used in the evaluation of segmentation algorithms, we can define a number of follow-up steps to collect a larger dataset. Preferably, this set is annotated by early music enthusiasts, as they have been exposed to many hours of listening to the genre and could be regarded as lay experts in the area.

We calculated boundaries using annotations by 22 participants. To make the annotation process more efficient, we need to investigate what the optimum number of annotations per composition is to get a reliable set of boundaries. The activity analysis approach described in [25] could supply further statistical underpinning of our conclusions, provided the approach is adapted to suit the scoring method used in our experiment.

The works need to be selected from a wider time span to make sure that the results are more widely generalisable. For the same reason, the selected works need to include instrumental music. To understand the impact of the performance, we need to extend the experiment with different performances of the same works.

The algorithms were tested with out-of-the-box settings: optimising the settings to late Renaissance polyphony could yield better results. Concerning algorithm design, it seems that much can be gained by taking the polyphonic structures such as overlapping voices into account. Whether existing algorithms can be adapted for this or new algorithms need to be developed is an open question.

Once the performance of boundary detection algorithms has reached an acceptable level, we will be able to create more high-level analytical procedures that support our goal of understanding tonal development in early music.

Acknowledgments

The authors would like to thank the participants of the experiment, Libio Goncalves Braz for reviewing the code, and the anonymous reviewers for their feedback.

8. REFERENCES

[1] C. Weiß, "Investigating style evolution of Western classical music: A computational approach," *Musicae Scientiae*, vol. 23, pp. 486–507, 2019.

[2] J. Yust, "Stylistic information in pitch-class distributions," *J. of New Music Research*, vol. 48, no. 3, pp. 217–231, 2019.

[3] D. Harasim, F. C. Moss, M. Ramirez, and M. Rohrmeier, "Exploring the foundations of tonality: statistical cognitive modeling of modes in the history of Western classical music," *Humanities and Social Sciences Communications*, vol. 8, pp. 1–11, 2021.

[4] M. K. Long, *Hearing Homophony: Tonal Expectation at the Turn of the Seventeenth Century*. Oxford University Press, USA, 2020.

[5] N. Cook, *Beyond the Score: Music as Performance*. London, U.K.: Oxford University Press, 2013.

[6] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *2000 Proc. of the IEEE Int. Conf. on Multimedia and Expo*, New York, NY, USA, 2000, pp. 452–455.

[7] O. Nieto and T. Jehan, "Convex non-negative matrix factorization for automatic music structure identification," in *2013 IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Vancouver, BC, Canada, pp. 236–240.

[8] B. McFee and D. P. Ellis, "Learning to segment songs with ordinal linear discriminant analysis," in *2014 IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Florence, Italy, pp. 5197–5201.

[9] B. McFee and D. Ellis, "Analyzing song structure with spectral clustering." in *Proc. of the Int. Soc. for Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014, pp. 405–410.

[10] J. Serra, M. Müller, P. Grosche, and J. L. Arcos, "Unsupervised detection of music boundaries by time series structure features," in *Proc. of the AAAI Conf. on Artificial Intelligence*, vol. 26, no. 1, 2012, pp. 1613–1619.

[11] C. Wang and G. J. Mysore, "Structural segmentation with the variable markov oracle and boundary adjustment," in *2016 IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, Shanghai, China, pp. 291–295.

[12] F. Fitch, *Renaissance Polyphony*. New York, NY, USA: Cambridge University Press, 2020.

[13] G. Zarlino, *The Art of Counterpoint: Part Three of Le Institutioni Harmoniche, 1558*. Translated by Guy A. Marco and Clade V. Palisca. Music Translation Series. New Haven, CT: Yale University Press, 1968.

[14] H. H. Eggebrecht and F. Reckow, "Kadenz," in *Das Handwörterbuch der Musikalischen Terminologie*. JSTOR, 1968.

[15] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research." in *Proc. of the Int. Soc. for Music Information Retrieval (ISMIR)*, New York, NY, USA, 2016, pp. 547–553.

[16] J. Paulus, M. Müller, and A. Klapuri, "Audio-based music structure analysis." in *Proc. of the Int. Soc. for Music Information Retrieval (ISMIR)*, Utrecht, The Netherlands, 2010, pp. 625–636.

[17] O. Nieto, G. J. Mysore, C. Wang, J. B. Smith, J. Schlüter, T. Grill, and B. McFee, "Audio-based music structure analysis: Current trends, open challenges, and applications," *Transactions of the Int. Soc. for Music Information Retrieval (ISMIR)*, vol. 3, no. 1, 2020.

[18] I. Deliege, "Grouping conditions in listening to music: An approach to Lerdahl & Jackendoff's grouping preference rules," *Music Perception*, vol. 4, no. 4, pp. 325–359, 1987.

[19] E. F. Clarke and C. L. Krumhansl, "Perceiving musical time," *Music Perception*, vol. 7, no. 3, pp. 213–251, 1990.

[20] F. Wiering, J. de Nooijer, A. Volk, and H. J. Tabachneck-Schijf, "Cognition-based segmentation for music information retrieval systems," *J. of New Music Research*, vol. 38, no. 2, pp. 139–154, 2009.

[21] D. Tomašević, S. Wells, I. Y. Ren, A. Volk, and M. Pesek, "Exploring annotations for musical pattern discovery gathered with digital annotation tools," *J. of Math. and Music*, vol. 15, no. 2, pp. 194–207, 2021.

[22] D. Bedoya, "Capturing musical prosody through interactive audio/visual annotations," Ph.D. dissertation, Institut de Recherche et Coordination Acoustique / Musique (IRCAM), Paris, France, 2023.

[23] D. Bedoya, L. Fyfe, and E. Chew, "A perceiver-centered approach for representing and annotating prosodic functions in performed music," *Frontiers in Psychology*, vol. 13, 2022.

[24] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart, "The musicality of non-musicians: An index for assessing musical sophistication in the general population," *PloS ONE*, vol. 9, no. 2, p. e89642, 2014.

[25] F. Upham and S. McAdams, "Activity analysis and coordination in continuous responses to music," *Music Perception: An Interdisciplinary Journal*, vol. 35, no. 3, pp. 253–294, 2018.

[26] B. L. Sturm and A. Flexer, "Validity in music information research experiments," *arXiv preprint arXiv:2301.01578*, 2023.

[27] C. F. Jekel, G. Venter, M. P. Venter, N. Stander, and R. T. Haftka, "Similarity measures for identifying material parameters from hysteresis loops using inverse analysis," *Int. J. of Material Forming*, vol. 12, pp. 355–378, 2019.

[28] D. Müllner, "Modern hierarchical, agglomerative clustering algorithms," *arXiv preprint arXiv:1109.2378*, 2011.

[29] P. Du, W. A. Kibbe, and S. M. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, no. 17, pp. 2059–2065, 2006.

[30] D. Turnbull, G. R. Lanckriet, E. Pampalk, and M. Goto, "A supervised approach for detecting boundaries in music using difference features and boosting." in *Proc. of the Int. Soc. for Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007, pp. 51–54.

[31] E. Bigand, "More about the musical expertise of musically untrained listeners," *Ann. of the New York Academy of Sciences*, vol. 999, no. 1, pp. 304–312, 2003.

[32] M. Phillips, A. J. Stewart, J. M. Wilcoxson, L. A. Jones, E. Howard, P. Willcox, M. du Sautoy, and D. De Roure, "What determines the perception of segmentation in contemporary music?" *Frontiers in Psychology*, vol. 11, p. 1001, 2020.

EXPLORING PERDE-SPACE: 3-D TONNETZ VISUALIZATION FOR MAKAM MUSIC

Recep Gül[1], Ozan Baysal[2], Yavuz Buruk[3],Yusuf Can Şeftali[4],
[1,2,]Istanbul Technical University, Turkish Music State Conservatory
[3,4,]Graduate School, Istanbul Technical University
recepgul, ozanbaysal,buruk18, seftali@itu.edu.tr

ABSTRACT

This paper introduces a similarity and weight metric designed to assess the distance between pitches and melodic fragments within the framework of Turkish Makam Music. Our approach hinges on adapting the Riemannian Tonnetz to represent pitch-based elements, termed "perde" (pitch), or "çeşni" (tetrachordal material for melodic fragments), within a three-dimensional vector space tailored specifically for Turkish Makam Music. We demonstrate the efficacy of this space as a reliable approximation for gauging similarity between musical objects and estimating the cost associated with traversing the music network using different distance metrics namely Euclidean, Minkowski, Manhattan and Cosine, thereby facilitating the construction of a weighted version thereof. Taking weighted relationships into consideration, this metric enables the computation of the cost associated with potential paths between musical objects and furnishes a new predictive tool for supervised learning models. Following the exposition of the geometric model and calculation methodology, we undertake a comparative analysis with results derived from a sample annotated by expert practitioners, revealing a notably high degree of correlation between our algorithm utilizing cosine distance and expert ratings, Spearman's rho = 0.828 with p-value < .001.

1. MOTIVATION AND RESEARCH

This paper aims to investigate the feasibility of mapping the pitch and melodic components, namely "*perde*" and "*çeşni*," within Turkish Makam Music onto a geometric or topological framework. This framework could serve as a basis for quantifying distance and similarity among these elements. The impetus for this inquiry emerged from Project makamNetz, which endeavors to depict all structures of "*makam*" and "*çeşni*," along with their interrelations, as an extensive network or map. Upon scrutinizing the topological configurations of this network, it became apparent that the relationships themselves do not possess uniform significance. This concern became particularly evident when delineating paths between different nodes, as the network assigned equal weight to paths of equivalent total distance, despite their varying frequencies in practical application.

Copyright: © 2024 Recep Gül et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

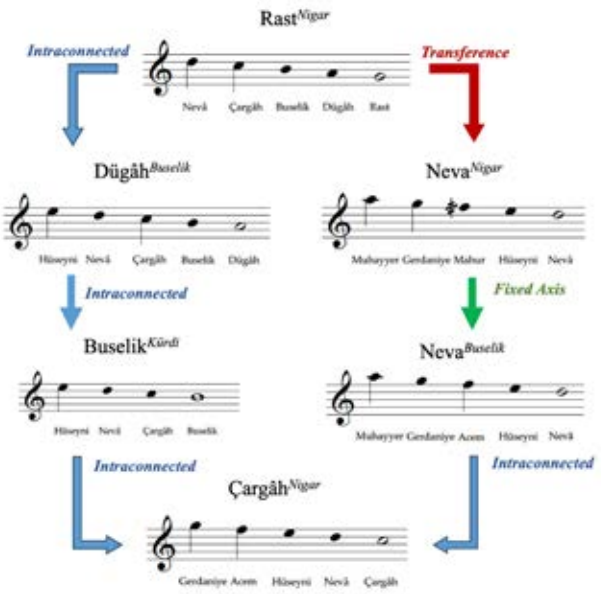


Figure 1. Two melodic progressions from Turkish makam music.

The makam space can be understood as comprising different *çeşni* elements, referred to as CMAKAM. Each element in this space consists of 3 to 5 pitches or "*perde*." These melodic structures evolve according to specific relational principles. Figure 1 depicts two configurations of Perde^{Çeşni} progressions outlining the structural body of two melodic lines . while specifying the *perde* collection from which various melodic structures can emerge. Both melodic lines begin and end on the same spaces, commencing the melodic progression with Rast^{Nigar} and concluding with Çargah^{Nigar}. However, the paths they take to reach their destination differ. The first progression (intraconnected) illustrates a smoother transition achieved by adding new notes to the existing melodic formula or by shifting the tonic of the tetrachordal or pentachordal structure. The second progression, though more unconventional, is still feasible. It involves transferring pitches entirely to a new axis—a fifth or fourth related to the original sequence (transference, indicated by red arrows)—or simply altering some scale degrees on the same axis (fixed axis, indicated by green arrows). From a progressional standpoint, relationships involving common tones or direct

transpositions are considered natural and are frequently encountered within various makams.

However, progressions utilizing varying *perde* material that share the same axis—which represent modal shifts—have not been previously modeled within Turkish Makam Music context and present a significant challenge. Without a proper hierarchical measurement principle, the two progressions depicted below exhibit similar topology and are considered equivalent in terms of distance, which is inaccurate from both practical and cognitive perspectives. Our aim is to model this differentiation, thereby creating a weighted network where different progressions carry distinct weights, reflecting varying path distances.

2. DISTANCE IN MUSICAL SPACE

The question of distance has garnered widespread recognition and exploration in music theory, with various spatial representations offered for different musical styles, particularly within the realm of common practice tonality. While some theorists grounded their assertions in music-theoretical principles, others rooted their work in cognitive approaches. Though this paper does not aim to provide an exhaustive survey of these approaches, it is pertinent to reference those that informed this study. [1] presents a hierarchical model of pitch relationships within a tonal context, which is supported by the cognitive research of Deutsch and Feroe [2]. This model investigates pitch distance across five distinct levels: octave, fifth, third, diatonic, and chromatic. Drawing from similar conceptual foundations, Lerdahl establishes a chordal space, wherein chords are interconnected through alternation PL transformations along the horizontal axis and major thirds along the vertical axis. In these representations, octaves, fifths, and thirds are accorded priority as the primary constituents of both pitch and chordal spaces.

Alternative approaches to measuring distance have also been proposed by Tymoczko, [3] in his seminal work "A Geometry of Music," where he delved into alternative pathways between scales, chordal objects, and voice-leading schemas. Tymocko's work predominantly relies on geometric representations of musical entities as nodes and links, or some variation of vector space. Both Tymocko and Lerdahl contend that distance in music constitutes a cognitive phenomenon and can be geometrically modeled, allowing for specific calculations. The inquiry into distance extends beyond the confines of music theory. A recent study employing network analysis of 456 annotated jazz improvisation transcriptions, coupled with cognitive experiments, revealed that perceived distance, as a measure of similarity, aligns with network distance measures (Merseal et al., [4]).

3. MAKAM PERDE SPACE

Prioritizing fifths and thirds is not unique to Western tonal context. A similar framework is applicable in Turkish Makam Music when deriving "*perde*" relationships, thus constructing a "Makam Perde" space or PMAKAM [1]. Figure 2 [2] depicts a section of the makam/perde space as a 2D Tonnetz representation. The horizontal dimension illustrates fifth, while verticality denotes third relations. In Turkish makam music, thirds and fifths are tuned according to the principles of the harmonic series. However, attempting to construct Turkish *perde* space based on these principles results in an impractically large conceptual Tonnetz where many closely related "*perde*" are distantly spaced. This is because constructing the vertical dimension based on natural thirds would yield "*perde*" tunings that are not practically used. Consequently, the result is a large horizontal Tonnetz that cannot form a cycle.

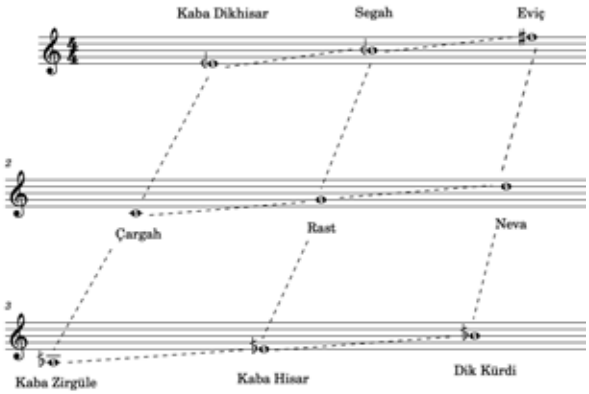


Figure 2. Section of makam perde space in 2d Tonnetz.

To address this question, we constructed a tonnetz without assuming octave equivalence, thereby contributing to the generation of "perde" space. The resulting geometry is depicted in Figure 3. According to this figure, Turkish Makam *Perde* Space (PMAKAM) is conceptualized as five fundamental layers interconnected via octave relations, as indicated by vertical slashed lines. It's worth noting that the upper-left and lower-right layers are also connected via octave relations, assuming a toroidal shape. Essentially, this space indicates the distance between makam objects in terms of thirds, fifths, or octaves. For example, on the third layer in the middle, we observe two frequently used "*perde*" called "*Rast*" and "*Neva*." The former is a fifth above the latter, positioned one step to the right on the x-axis. Similarly, "*Segâh*" is a natural third above "*Rast*," so it is one step up on the y-axis. "*Gerdaniye*," connected to "*Rast*" via a slashed line, is an octave above, shifting the layer one step up on the z-axis, thus serving as special connectors between layers. It's important to note that the tonnetz representation does not refer to static frequencies but rather denotes third, octave, and fifth relations, historically the building blocks of Turkish Makam Space.

[1] When naming PMAKAM and CMAKAM we follow similar naming convention to Hook [5], an important inspiration in formulating the spatial and topological properties of the space.

[2] This representation is presented in TUMAC BSE No.25: "Multi-dimensionality in Makam Musics" by Dr. Michael Ellison `https://www.youtube.com/watch?v=6-SSscA8ZRc`

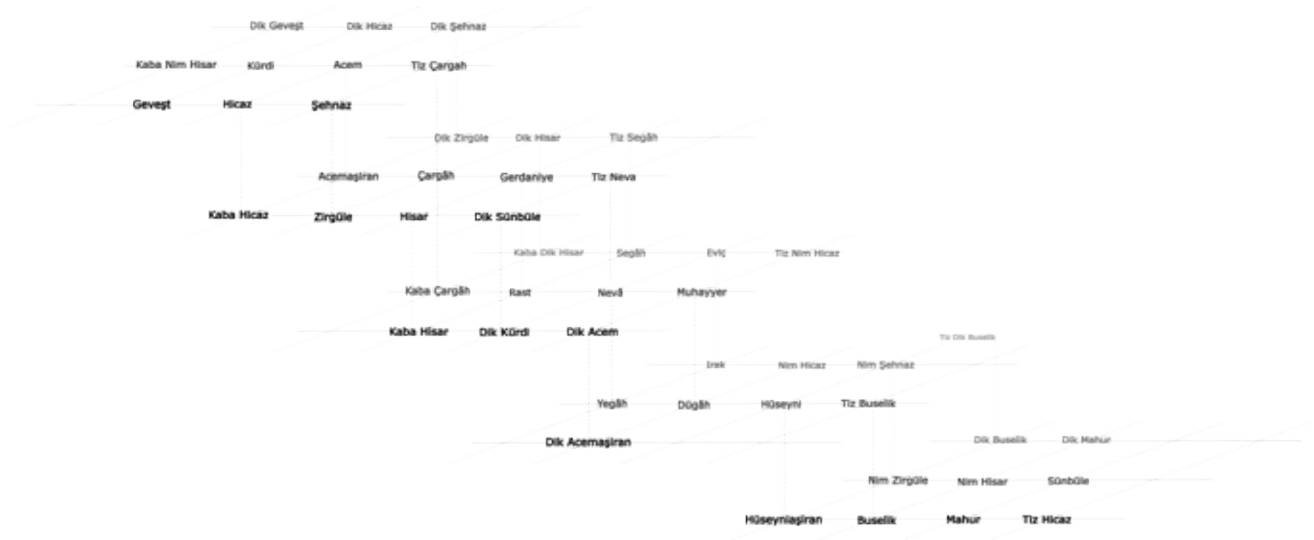


Figure 3. Three Dimensional Makam Tonnetz.

4. MODELING AND ALGORITHM

Related to our first research question we discovered that the 3D tonnetz model of PMAKAM can be used to calculate distance between different cesnis particularly the ones that has primary axis relation. In this model a perde is a vector in R^3 and a cesni is 3 by n matrix composed of 3 to 5 perdes. In this space we utilized different distance calculation methods and compared the results with the survey results we obtained from 16 expert raters. We have used 3 different algorithms to calculate the distance between 2 sequences and used the average of these 3 algorithms as the final distance measurement between these 2 sequences. In the 1st algorithm we measured the distance between each perde in both sequences and these distances are cumulated to give the net distance. If one of the sequences has fewer perdes compared to the other one, then the missing perdes are substituted by the starting perde. Formally we can define the process as below:

$$P_{axis} = [0, 0, 0]$$
$$PTonnetz = \{perde \mid perde = [x, y, z]^T - P_{axis}^T \quad (1)$$
$$, x, y, z \in \mathbb{Z}\}$$

Thus we can state that $PTonnetz$ is a geometric representation of a conceptual perde space whose structure depends on primary axis perde situated on P_{axis}. The resulting space is a subspace of $\mathbb{Z}^3$. Next we define a two cesni sequence to compare:

$$Cesni_1 \in PTonnetz^{1xn} \text{ and } Cesni_2 \in PTonnetz^{1xm}$$
$$CostFunction = \{Euclidean, Minkovski, Manhattan, Cosine\}$$
$$Distance(C_1, C_2) = \sum_{i=1}^{k} CostFunction(x_i, y_i)$$
$$\text{where } x_i \in C_1, y_i \in C_2 \text{ and } k = max(m, n) \quad (2)$$

We can now define the algorithms to calculate distance. The algorithm 1 uses the following formula:

```
Step 1. if n > m:
    extraindex = {i for m < i <= n}
    for j in extraindex:
        Cesni_1[j] = {P_axis}
Step 2. elif
    extraindex = {i for n < i <= m}
    for j in extraindex:
        Cesni_2[j] = [P_axis]

Step 3. distance = Distance(Cesni_1,
    &Cesni_2)
```

The missing perdes are substituted by a randomly selected perde within the sequence in the 2nd algorithm. Thus we simply replace the code in step 1 and 2 that assigns the extra index of Cesnis to P_{axis} with random element from $PTonnetz$.

The 3rd algorithm calculates all pairwise distances within the sequence and cumulates them. The distances in the 3-dimensional space are calculated with 4 different distance calculations: Euclidean, Manhattan, Cosine and Minkowski (p = 10) methods. In the 3rd algorithm, the minimum of these 4 distance measurements is taken into account. Since it is simpler to show mathematically we can formalize it as below:

$$Distance(C_1, C_2) =$$
$$min(\sum_{i=1}^{n}\sum_{j=1}^{m} CostFunction(x_i, y_j)), x_i \in C_1, y_j \in C_2 \quad (3)$$

We also have to note that in all algorithms after calculating all pairs in question we scale the cost function to a value between 1 and 5 matching maximum cost to 5 and minimum to 1. This helps us to compare and visualize the results with the experiment we obtained from the participants and formalize the third algorithm. [3] .

[3] Detailed computations can be accessed via GitHub (https://github.com/yavuzburuk/makamnetz)

Figure 4. Experts' Evaluations versus Distance Functions.

5. EXPERIMENT RESULTS WITH EXPERTS

The experiment engaged sixteen expert raters, all possessing formal backgrounds in Turkish music conservatory education and aged between 25 and 45 years old. The raters represented a diverse group including university faculty members, Ph.D. and M.A. students, as well as professional practitioners. Each rater was presented with a list of 36 Perde^{Çeşni} changes, encompassing fixed-axis transformations within Turkish music theory, such as transitions from Dügah^{Hicaz} to Dügah^{Uşşak} or Dügah^{Saba} to Dügah^{Buselik} ⋯ etc. Their task involved rating the perceived difficulty of executing each Perde^{Çeşni} change on a scale ranging from 1 to 5. Ratings of 1 indicated the smoothest and easiest transitions, while ratings of 5 indicated the most demanding and challenging transitions.

Subsequently, the ratings provided by the expert raters underwent inter-rater reliability analysis using Krippendorff's Alpha. The resulting Krippendorff's Alpha value, K-Alpha = 0.3846, suggested a moderate level of agreement among the raters. This indicated that while there was some degree of consensus among the raters, variations in interpretation existed, likely stemming from differences in individual strategies when assessing the Perde^{Çeşni} changes. However, additional analysis uncovered a substantial level of agreement among the expert raters when evaluating the Perde^{Çeşni} changes with the lowest five and highest five means, yielding a Krippendorff's Alpha value of 0.720. This finding supports the notion of a "gray area" regarding the transition cost of some Perde^{Çeşni} changes, influenced by personal strategies derived from the experiences of musicians.

Additionally, we evaluated our algorithm's performance by comparing it with expert ratings using four different distance functions: Cosine, Minkowski, Manhattan, and Euclidean distances can be observed from Figure 4. Spearman correlation tests revealed significant and positive correlations between our algorithm and the expert results, Cosine distance giving the highest correlation of 0.828 with p <.001 (Table 1).

These results demonstrate strong and statistically significant correlations between our algorithm's predictions and expert judgments, across various distance functions, indicating the effectiveness of our approach in predicting the perceived difficulty of executing the Perde^{Çeşni} changes.

		Expert Rating
Cosine	Spearman's rho	0.828
	p-value	<.001
Minkowski	Spearman's rho	0.695
	p-value	<.001
Manhattan	Spearman's rho	0.659
	p-value	<.001
Euclidean	Spearman's rho	0.662
	p-value	<.001

Table 1. Spearman Correlation Tests: Experts' Evaluations versus Distance Functions.

6. CONCLUSION

This paper contributes to the understanding and quantification of distance and similarity among perde and çeşni components within Turkish Makam Music. Our investigation revealed the inadequacy of existing models in capturing the nuanced differences in distance within Makam music, particularly concerning modal shifts. By drawing from established music theory principles and cognitive approaches to distance measurement, we developed a novel approach rooted in geometric representations, specifically the Tonnetz model. Through the construction of this framework, we aimed to address the challenge of accurately representing the varying significance of relationships between different melodic progressions.

Octave equivalence poses a unique challenge in Turkish music, as perde names vary according to registral differences (e.g., *Yegâh-Nevâ-Tiz Nevâ, Rast-Gerdaniye, Dügâh-Muhayyer*). This variability complicates the application of standard tonnetz frameworks, which typically rely on fifths and thirds. To address this, we extended the model to incorporate octave relations, thus devising a Makam Perde Space (PMAKAM) represented as a 3-D Tonnetz. This enhancement allows us to elucidate the distance between Perde^{Çeşni} structures in terms of thirds, fifths, and octaves, providing a more accurate depiction of Makam music's structural relationships.

Our algorithmic approach facilitated the calculation of distances between different *Çeşni* structures, particularly those sharing primary axis relations. By comparing these results with expert ratings, we demonstrated the efficacy

of our algorithm in predicting the perceived difficulty of executing Perde^{Çeşni} transformations.

However, our study is not without limitations. Notably, we treated "perde" as fixed points in space, neglecting their movable quality within a frequency bandwidth. This oversimplification may limit the applicability of our findings in contexts where "perde" exhibit greater flexibility. Additionally, our focus was primarily on a specific subset of Perde^{Çeşni} transformations within Turkish Makam Music, particularly those based on the same perde axis (fixed-axis). Future research should broaden this scope to encompass a wider range of transformations, including intra-connected and transference relations. Lastly, it is important to note that Makam music encompasses a rich array of expressive elements beyond pitch, including rhythm, ornamentation, and timbre, which were not fully addressed in our study. All these factors might possibly contribute to subjective variability in musical evaluation leading to the moderate level of agreement among expert raters in the "gray area." Future research could integrate these aspects to create a more holistic representation of Makam music, allowing for a deeper understanding of its expressive nuances.

By combining theoretical insights with computational approaches, we pave the way for a more comprehensive understanding of the intricate relationships within Turkısh Makam Music, thereby enriching both theoretical discourse and practical applications in musicology and computational music analysis.

Acknowledgments

This work is supported by the Scientific and Technological Research Council of Turkey, TÜBİTAK, Grant 122K923.

7. REFERENCES

[1] F. Lerdahl, *Tonal pitch space*. Oxford University Press, USA, 2001.

[2] D. Deutsch and J. Feroe, "The internal representation of pitch sequences in tonal music," *Psychological Review*, vol. 88, pp. 503–522, 11 1981.

[3] D. Tymoczko, *A Geometry of Music: Harmony and Counterpoint in the Extended Common Practice*. Oxford University Press, 2011.

[4] H. Merseal, R. Beaty, Y. Kenett, J. Lloyd-Cox, de Manzano, and M. Norgaard, "Representing melodic relationships using network science," *Cognition*, vol. 233, p. 105362, 2023.

[5] J. Hook, "Exploring Musical Spaces: A Synthesis of Mathematical Approaches." Oxford University Press, 02 2023.

EXPLORING IMMERSION AND SPATIAL AUDIO PERCEPTION IN CONCERTS FOR HEARING-IMPAIRED AND NORMAL HEARING AUDIENCES

Jesper ANDERSEN[1] and **Stefania SERAFIN**[2]

[1]**The Royal Danish Academy of Music**, Copenhagen, Denmark
[2]**Aalborg University Copenhagen**, Copenhagen, Denmark

ABSTRACT

In this paper we present the results of an investigation performed during a concert designed specifically for individuals with a hearing impairment. We remixed a popular song for the Ambisonic concert hall at The Royal Danish Academy of Music. 23 participants at the concert filled out an immersive and spatial audio perception questionnaire and seven of them participated in individual interviews. Results show that spatial audio perception is better for individuals with normal hearing as opposed to individuals with hearing impairment, but there is no significant difference regarding the sense of immersion. The investigation also underlines the challenges of running scientific experiments in concert settings.

1. INTRODUCTION

With devices like cochlear implants (CI) and hearing aids (HA), most hearing impaired (HI) individuals can achieve good speech intelligibility, whereas music perception remains problematic [1]. However, most HI report that music continues to be a part of their life and that they respond emotionally to music [2].

The fact that longer reverb time and echoey rooms affects music enjoyment for HI negatively [3] [4] makes concertgoing problematic. And since instrument discrimination is difficult for HI, the auditory complexity of a concert experience can lead to a decreased music enjoyment [5]. The hearing impairment and the general social aspects of concertgoing can be difficult to overcome, which means that many HI tend to avoid music listening other than at home [6].

A lot of research has investigated how to improve music perception for HI [7], but there has only been a few studies of concert attendance [8]. Since concertgoing and listening together is an important factor of the general quality of life [9], this is an area that begs further research. It was shown that a widely spread spatialization of speech and noise improves speech intelligibility for HI [10]. So, it could be speculated that an immersive reproduction of music could impart a higher degree of instrument segregation and enjoyment for HI, especially in a concert setting.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

In this paper, we investigated how hearing impaired individuals experience immersive music delivered as part of a concert experience.

Specifically, the research presented here is placed at the intersection of music appreciation for hearing impaired individuals, immersion studies and concert research. From the perspective of music appreciation, we seek a better understanding of how individuals with hearing impairment experience music during a social event such as a concert. From the perspective of immersion studies, we aim to better understand the concept of immersion and how it can be measured for hearing impaired individuals. From the perspective of concert research, this is an emerging topic discussed in the next section.

1.1 Concert research

In recent years, there has been a trend to increase the ecological validity of empirical work in music cognition research [11, 12]. One of the areas where this trend has been seen is in the combination of concerts and scientific activities [13, 14]. The empirical methods used in such settings range from self-reports, sometimes continuous [15], measurement of inter-subject correlations [16], and capturing motion through mobile phones [17]. Another interesting research direction is the study of the effect of room acoustics in the emotional response to music [18].

In some research institutions, sophisticated performance- or concert halls specialized for concert research have been created, such as the Livelab at McMaster University in Hamilton, Canada [1] and the ArtLab at the Max Planck Institute for Empirical Aesthetics in Frankfurt, Germany [2].

In [8], the authors describe the execution and evaluation of a concert specifically designed for individuals with CI. In particular, six pieces were commissioned with a focus on individuals with CIs. The concert was evaluated through focus group interviews. In the interviews, spatialization was identified as an important element by both normal hearing and individuals with hearing impairment. The paper also presented some considerations regarding the familiarity of the music the audience was exposed to, and whether they could develop appreciation through increased exposure. It is well known from the literature that the familiarity of music is important to increase its enjoyment [19] especially for the HI [20].

In [21], a research program is proposed to evaluate music

[1] https://livelab.mcmaster.ca
[2] https://www.ae.mpg.de/artlab/information.html

listening in classical concerts. The paper acknowledges how concerts and concert listening experiences have already been acknowledged as worthwhile research topics by a multitude of disciplines. However, a more thorough, systematic, and transdisciplinary research is still needed. One important aspect stressed in the paper is that concert curators need to form an essential part of a research team, since the research inquiries need to fit as part of the artistic experience. Moreover, it is pointed out how a research program that investigates the concert frame and its components can only be performed in interdisciplinary teams.

Overall, as stated in [22], concerts can be considered a presentational musical field, where participants are divided into performers and audience, as opposed to the participatory musical field, where participants move between being performers and being audiences. This means that concerts are an interesting venue of research where ecological validity is maintained, but with the advantage that the roles of the individuals involved are rather well defined.

1.2 Measuring immersion

Researchers from different domains, from virtual reality to games and to musical experiences proposed several definitions of the term immersion and how to measure it.

In the field of virtual reality, one of the prominent researchers, Mel Slater, defined 'immersion' as an objective property of a system. Higher or lower immersion depends on the extent to which a virtual reality system can support natural sensorimotor contingencies for perception [23, 24].

In [25], Hyunkook Lee examines the definitions of various concepts related to immersion and integrates them into a general conceptual model of immersive experience. The paper considers the concept of immersion as well as related concepts such as presence and involvement, in order to create some rigor and organization of the existing definitions proposed in the literature.

An attempt to define immersion in the context of audio-visual experiences is also provided in [26]. Taking into consideration the multitude of research and definitions regarding immersion, the authors define the term as *a phenomenon experienced by an individual when they are in a state of deep mental involvement in which their cognitive processes (with or without sensory stimulation) cause a shift in their attentional state such that one may experience disassociation from the awareness of the physical world.*

In the context of musical experiences or audio-visual experiences, most research examines the influence of sound delivery methods in enhancing immersion [27, 28]. Such explorations are mostly performed in laboratory settings, with one subject at the time, and with the different conditions randomized.

In [29] (part 1 and part 2) different spatial audio reproduction methods were evaluated and it was found that expert listeners and inexperienced listeners use very different languages to describe the sense of immersion.

When looking at methods to measure immersion, Rumsey observes how there has been a unwillingness to break free from the concept of reference stimuli [30]. Rumsey states that there may be advantages to adopting some form of reference-free evaluation. He observes that there is still the open question of defining immersion carefully enough, pointing at the different attempts undertaken. He also mentions that non-experienced listeners may not have the same preference patterns as experienced ones in this context, and that additional low-slung loudspeakers can improve the results in certain cases.

In [31], immersion is defined as a psychological state characterized by perceiving oneself to be enveloped by, included in, and interacting with an environment that provides a continuous stream of stimuli and experiences.

In order to measure immersion in musical settings, a recent questionnaire has been developed [31]. The questionnaire was developed to understand whether there is a relationship between the increasing spatiality of sounds and the listener's emotional response.

In this paper, by using an adaptation of the questionnaire presented in [31] and [32], we investigated immersive music experiences in a concert setting and understanding whether immersion and spatial characteristics perceived differently by hearing impaired and normal hearing users.

2. CONTEXT OF THE RESEARCH

This research is a part of a larger cooperation between the Multisensory Experience Lab at Aalborg University [3], The Royal Danish Academy of Music (RDAM) [4] and Copenhagen Hearing and Balance Center (CHBC) at Rigshospitalet [5]. Through concerts and other events, music for hearing impaired is investigated.

This paper is based on a concert-event that took place on October 1st 2023 in New Hall at RDAM. For the concert, a band (drums, electric bass, grand piano and voice) played eight songs (*What a wonderful world, Lyse nætter, Jeg har elsket dig så længe jeg kan mindes, To lys på et bord, Imagine, Den gamle skærslippers forårssang, Forårsdag* and *Papirsklip*). The repertoire was chosen to put together a programme of familiar songs with a broad appeal for the Danish audience, which was expected to have a high average age (at the event the average age was 57,6 years).

In order to help the audience feel comfortable, the event started with a sing-along song. This is in line with the Danish tradition of singing together at concerts and social events. The event also ended with a sing-along song. The first sing-along song was *Septembers himmel er så blå*, and the final one *Nu falmer skoven trindt om land*. Both songs are very familiar to a Danish audience.

The concert also involved a test of melodic contour identification (MCI) [33], where participants were asked to identify different melodic contours (rising, falling or arching). There was also tryouts of a haptic bench, which is a bench with footrest, where vibrating actuators are mounted and configured to support the listening experience of the voice and bass in the music.

[3] https://melcph.create.aau.dk

[4] www.dkdm.dk

[5] https://www.rigshospitalet.dk/english/departments/centre-of-head-and-orthopaedics/department-of-otorhinolaryngology-head-and-neck-surgery-and-audiology/Centres-and-Units/Pages/Copenhagen-Hearing-and-Balance-Centre.aspx

Figure 1. Image from the concert

To investigate the sense of immersion for the audience, one song was played back through an Ambisonics system in the hall. The music played at the event, was a studio recording of a band playing the Danish folksong *Der er ingenting i verden så stille som sne* (There is nothing in the world as quiet as snow). It is a "live in the studio" multi-track recording with overdubbed lead/backing-vocals and a few extra instruments. The playback of the immersive song was in the programme after the fourth band-song.

Since the event had a strong focus on listening and was investigating listening capabilities, there were no assistive listening devices, texting or sign interpreter. Out of the approximately 45 people in the audience, 23 took part in the research, by filling out questionnaires in realtime.

3. DESIGN OF THE AMBISONIC HALL AT THE ROYAL DANISH ACADEMY OF MUSIC

The Ambisonics system in New Hall at RDAM was established in 2019. It has 44 loudspeakers (Dynaudio LYD5), arranged in a dome around the audience, and four subwoofers (Dynaudio SUB6) on the floor. (see Table 1, Figure 3 and Figure 4. All loudspeakers/sub's are fed from a DAD AX32 converter. The speakerdome is arranged in five rings with two loudspeakers in the top above the center.

Ring	Number	Height (cm)
Floor	1-8	40
Panel	9-20	160
Above panel	21-28	260
High	29-34	320
Top	35-38	360
Zenith	39-40	400
Subwoofers	41-44	Floor
Stereo pair	45-46	260
Extra front	47-48	40

Table 1. The speakers' configuration in the Ambisonic concert hall at The Royal Danish Academy of Music.

The loudspeakers are permanently installed in the hall and designing the setup required both perceptual, technical, practical and architectural considerations. For instance, it was not possible to mount loudspeakers at the floor even though it would have been optimal to ensure the possibility of placing sounds anchored at ground level.

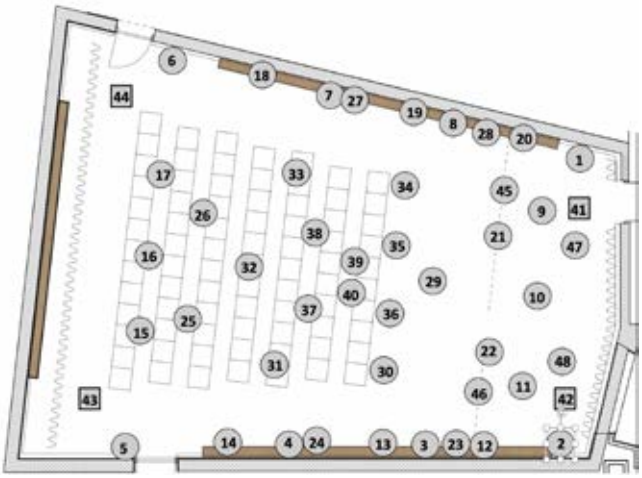


Figure 2. Speakers' placement in the hall. See the list in Table 1.

Figure 3. Speakers' placement in the hall. Side view.

The hall is trapezoidal and the ceiling not high enough to accommodate a perfect sphere, so loudspeakers are individually processed to mimic the optimal placement. This is achieved through delay, attenuation and equalization from the SPQ speaker management card in the DAD AX32.

New hall at RDAM is built for concert experiences rather than laboratory research. The acoustics are variable with an RT60 from 1,0s to 1,2s and designed for small classical ensembles. For this event the room was damped to an RT60 of 1,0s. Even though this is a short reverb for a concert space, the room acoustics are still an active player in the musical experience. From a scientific point of view, this increases the challenges because the distribution of sound is uneven according to where the listener is located in the hall. These differences especially play a role, when running experiments with a combined diverse audience.

The same room has been simulated visually and acoustically in [34], where a virtual reality simulation (VR) was proposed. This was done in order to provide virtual accessibility when the room was closed due to COVID-19 restrictions. In this case the sonic component is delivered through headphones with a custom made Ambisonic to binaural rendering engine.

4. METHODOLOGY

4.1 Mixing for the Ambisonic Room

For the audio mix, standard equalizing (FabFilter Q3) and dynamic compression (FabFilter C2) was applied to optimize the sound of the individual sources before placing them in the three dimensional space. To get a clearly focused placement, most instruments were played through only one single loudspeaker. To give the audience a clear immersive experience, it was chosen to create an enhanced spatial experience with the instruments located far apart in stead of recreating a naturalistic placement of the instruments on the front stage area. The whole mix was spread, as individual channel-based sources, across the 3D soundscape as shown in Fig. 5 and Fig. 6.

The audio-mix was played from an Apple MacMini running ProTools to the loudspeakers via a MADI-connection from an RME MadiFace to the DAD AX32 converter.

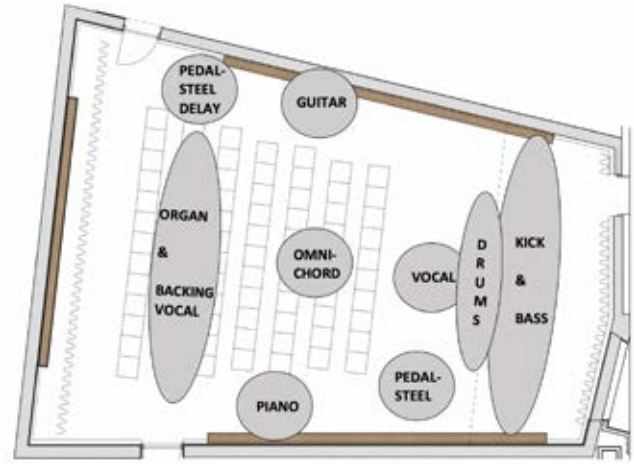


Figure 4. Placement of the sound sources in the immersive mix.

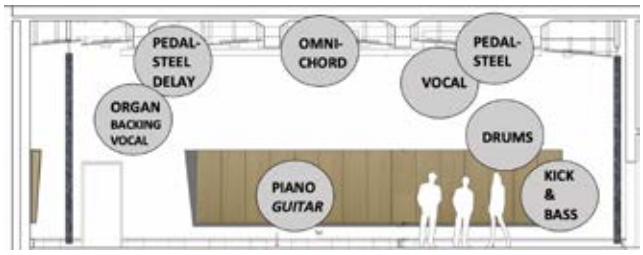


Figure 5. Placement of the sound sources in the immersive mix. Side view.

4.2 Questionnaire development

We prepared a questionnaire in order to measure how participants experienced the immersive mix, both from the perspective of the challenges usually experienced by HI, the general music appreciation and immersion characteristics. The questionnaire is based on a list of statements which the participants rated from 1 to 6 in relation to how strongly they disagreed or agreed.

The questions related to immersive characteristics were adapted from the immersive music questionnaire [31]. Questions related to music appreciation with a focus on challenge for hearing impaired were extracted from [32].

All questions were translated to Danish. We opted for a short questionnaire which is preferable, in many cases, since it avoids the adverse effects associated with an extended time of completion [35]. This is especially the case when the majority of the audience is represented by individuals with hearing impairment, that are known to be easily subject to fatigue [36]. We therefore limited the questionnaire to the 11 questions shown in the first column of Table 4. The questions related to music appreciation (questions 1-4), ability to discriminate instruments and their location (questions 5 and 6) and immersion (questions 7 to 11). The questionnaires were handed out on paper to the participants.

4.3 Interviews

One week after the concert, seven of the HI were interviewed individually about their experience. The interviewees are all a part of a larger study investigating the effect of music training on concert perception (awaiting submission). The interviews were carried out as semi-structured telephone interviews [37] [38], had an average duration of 20.5 minutes and concerned all the elements of the concert, including the immersive playback. All interviews were recorded and the analysis was carried out afterwards.

5. PARTICIPANTS AND THEIR HEARING PROFILE

The concert was an open public event. Of the approximately 45 individuals who participated only 23 data entries were considered eligible in the questionnaire. Some participants either did not answer more than one or two questions and some did not answer at all.

Participation was voluntary, and participants were not compensated. The recruitment process utilized various channels, including:

1. Danish CI social media groups.
2. Local and regional hearing associations (høreforening).
3. Local and regional centers for *Education and Communication* (Center for Specialundervisning for Voksne, Roskilde og København).
4. The *Center for Hearing and Balance* at the local hospital (Rigshospitalet).
5. Personal invitations.

While entering the concert hall, each participant was assigned an animal to replace their name, for respecting anonymity and asked to fill out the demographic informations as well as their hearing profile. They were also asked if they played or had played any musical instrument, how many concerts per year they attended, and to what degree they felt part of a shared musical experience when attending a concert. The replies can be found in Table 2.

Animal	Gender	Age	Hearing profile	Music	Concerts / year	Musical partaking
Alligator	F	35	2CI	No	1-2	some
Penguin	M	64	2CI	No	20	high
Hyena	F	66	2CI	no	5-8	high
Goat	M	68	2CI	no	4-5	some
Shrimp	F	?	2CI	piano (5 years)	10	high
Squirrel	F	74	1CI	guitar	1-2	some
Rhinoceros	F	74	1CI	no	0	some
Duck	M	67	1CI	no	0	not at all
Trout	M	78	1CI, 1 HA	clarinet	1	some
Bee	F	33	1CI, 1 HA	plays and sings	rarely	high
Raven	F	67	1CI, 1 HA	piano teacher	12-14	some
Rabbit	F	28	2 HA	no	one	some
Ladybug	F	26	2 HA	violin	5-10	high
Turkey	M	50	2 HA	guitar	0-1	some
Salmon	M	?	2 HA	no	10	some
Whale	F	78	N	klaver	2	high
Beaver	F	?	N	10	2	high
?	F	59	N		5	high
Bear	M	38	N	singing	2-3	high
Eagle	F	53	N		20	high
Scorpion	F	38	N	no	5-6	high
Snail	F	43	N		4	high
Panda	F	52	N		1-2	high

Table 2. The profile of the participants to the concert who filled the questionnaire described.

Profile	Number	Age (mean $\pm$ std)
2CI	5	58 ± 16
1CI	3	72 ± 4
1CI,1HA	3	59 ± 23
2HA	4	35 ± 13
NH	8	52 ± 14

Table 3. Summary of the profiles of the participants to the concert who filled the questionnaire described.

6. RESULTS

Table 4 shows the results of the answers to the questionnaire question by question in terms of mean and standard deviation for hearing impaired (column 2) and normal hearing listeners (column 3).

6.1 Analysis of Questionnaire

For analysis, we divided the participants in a group of hearing impaired individuals (n= 15), and a group of normal hearing (n= 8). The hearing impaired group included individuals with two cochlear implants (2 CI), one cochlear implant and deaf on the other ear (1 CI), one cochlear implant and one hearing aid (1 CI, 1 HA), and two hearing aids (2 HA) (see Table 3).

The mean and standard deviation of the answers from all participants can be seen in Table 5. From the table it can be noticed that the ratings by NH is generally higher for all the items to individuals with hearing impairment.

Due to the lack of normality, non-parametric tests were necessary for comparing item distributions in this dataset. A Wilcoxon rank-sum test showed a significant difference in the enjoyment of the music (question 1, $p < 0.05$), the appreciation of the overall soundscape (question 4, $p < 0.05$)), the ability to localize sounds (question 5, $p < 0.05$)), the ability to distinguish between different instruments (question 6, $p < 0.05$)), and the sense of being captivated by the listening experience (question 8, $p < 0.05$)). This is confirmed by the literature on hearing impaired individuals in social settings, where it is known that they have difficulties in localizing and distinguishing between many sound sources, also in musical settings [39].

Figure 6 visualizes the results of the questionnaire as boxplots. It can be seen that the HI had difficulties in perceiving directional cues and in instrument segregation (question 5-6), while they still enjoyed the playback (question 1-4). Also, the large deviation within the HI group compared to NH is clearly shown.

6.2 Analysis of Interviews

Interviewees all reported to have enjoyed the immersive playback and most of them felt somehow surrounded by sound, compared to listening to the band performance, which took place on the front stage.

The sensation of being surrounded by sound did, however, not lead to any perception of direction from the individual instruments. One single interviewee (a dual CI-user) reported very good perception of direction and instrument discrimination, which underlines the finding that there is a large deviation within the HI-group, while replies from NH is more homogeneous.

A CI+HA-user reported to have very good stereo-

Question	HI	NH
1) The piece was very enjoyable.	4.71 ± 1.37	5.87 ± 0.35
2) The melodies were very enjoyable.	5.07± 1.14	5.75 ± 0.46
3) The rhythms were very enjoyable.	5.03 ± 1.07	5.75 ± 0.46
4) The overall soundscape was very enjoyable.	4.92 ± 1.07	6 ± 0
5) I can tell where all the sounds were coming from.	3.00 ± 1.57	5.12 ± 0.83
6) I can distinguish different instruments.	4 ± 1.51	5.75 ± 0.46
7) I felt that the music surrounded me.	5.27± 1.19	5.87 ± 0.35
8) The listening experience captivated me.	4.8 ± 1.31	5.87 ± 0.35
9) My listening experience was similar to a live concert.	4.09 ± 1.44	5 ± 1.3
10) The music seemed detached from the loudspeakers.	4.09 ± 1.29	4.25 ± 1.388
11) The music resounded from everywhere.	4.63 ± 1.36	5.5 ± 0.75

Table 4. Questions asked to the participants and their mean and standard deviation (1-6 Likert scale) for hearing impaired (HI) and normal hearing (NH).

perception when streaming directly to the CI/HA for home-listening, while spatial perception in a real room was very bad. In spite of the live acoustics of the hall, interviewees of this study reported a high enjoyment of the music. Investigating the influence on concert hall acoustics on music enjoyment of HI is thus a relevant area of further studies.

7. DISCUSSION

Playing the immersive music production confirmed several challenges experienced by individuals with hearing impairment, such as the difficulty of segregating and localizing instruments. However, these results need to be considered with caution, given the relatively low number of participants that filled out the questionnaire.

Despite the relatively short questionnaire, some of the participants left some questions unanswered, reducing the amount of data collected. This is true both regarding their demographics, ability to play an instrument or even the questions regarding immersion.

It is worth noticing that the qualitative interviews that took place a week after the experience confirmed the data from the questionnaire.

Overall, we encountered different challenges when running scientific experiments during a concert experience. Especially when considering that the majority of the audience were hearing impaired and elderly individuals.

One challenge is the uncontrolled environment of a concert hall (as opposed to the controlled settings of a laboratory. Another challenge is how to balance between a scientific experiment and a musical experience pleasant for the audience. The audience was invited to a concert with a special focus on hearing impaired individuals. It was therefore just as important to focus on creating a concert experience as it was to have the scientific experiment as a goal.

It was also important to make sure that the scientific experiment proved beneficial to improve the quality of life of this particular population, and not a mere scientific inquiry that only satisfies the curiosity of the researchers.

For these reasons we kept the number of questions in the questionnaire rather limited (11 questions), selecting the ones we found most relevant to measure immersion and music appreciation. Concert research makes it hard

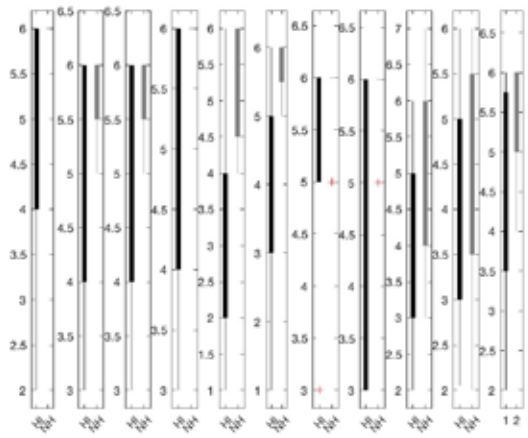


Figure 6. Box plots for the 11 questions for hearing impaired (HI) and normal hearing (NH). The boxplots clearly indicate the large deviation in the answers from the individuals with hearing impairment as opposed to those with normal hearing.

to perform traditional scientific experiments. For example, it could have been interesting to investigate the effect of different strategies of spatialization on immersion, or the overall effect of changing one variable in the spatialization. That would have required playing the same piece of music more than once in a randomized order. While this is a proper scientific methodology, it would have introduced a high level of fatigue in the audience involved. An alternative approach could have been to invite two groups of participants to two different concerts, but that would have required a much larger sample population in order to provide some statistically valid conclusions.

It would also have been interesting to triangulate the questionnaire's results and the qualitative interviews with captured objective data such as physiological data. This would have required some rather intrusive hardware and it is important to stress that the involved audience came to the event to listen to music, not to participate to a scientific experiment.

In future studies, we could consider other ways of data

collection, for example through mobile phones. This would require the development of a user friendly app suitable for the target audience. Data collection during concerts was done for example in [17], and the experience showed some limitations, despite addressing populations without reported impairments.

Although all individuals brought a smartphone, it would have required some additional technical support to use it as an additional measuring device.

It is well known that the listener's position, with respect to loudspeakers, creates precedence effect [40], nobody reported on that in the questionnaire. And only one participant reported in the interview, that the listening position had been close to the loudspeaker playing the guitar.

One item for further discussion is the choice of the music at the concert. As previously mentioned, in another concert for individuals with cochlear implants, some dedicated pieces of music were commissioned [41].

In the research presented in this paper we decided to choose a rather familiar song for different reasons. First of all, in previous focus group interviews it was clear that individuals with hearing impairment are interested in reaching a level of music appreciation which is similar to normal hearing for any kind of music, and they are not interested in having music specifically composed for their hearing devices. This is related to the importance of music as a social interaction and gathering, regardless of the ability of the listeners.

Individuals with hearing impairments do not want to be placed in an isolated category. Moreover, as previously mentioned, there is some previous evidence on the correlation between the familiarity of a musical experience and the appreciation [19], [20].

8. CONCLUSION

In this paper we described an experiment run as part of a concert experience for individuals with hearing impairments. The goal of the experiment was to investigate how individuals with hearing impairment experience immersive music compared to normal hearing individuals.

The results of the experiment confirm existing knowledge, for example the difficulty in localizing/discriminating instruments and that there is a large deviation in the listening capabilities of the HI. It is a positive sign that individuals with hearing impairment seem to enjoy music almost as much as the normal hearing.

The experiment showed several methodological challenges involved when running ecologically valid experiments. For example, the listening experience took place in a concert hall as opposed to a traditional laboratory setting. The audience at the concert included mostly elderly individuals with hearing impairments, which introduced additional challenges when measuring their experiences.

In the future, we plan to investigate other methodologies for measuring audience experience at similar events, for example combining questionnaires and interviews with physiological data such as heart rate and motion measurements.

9. ACKNOWLEDGMENT

We would like to thank the participants at the concert and Lone Marianne Percy-Smith from the Copenhagen Hearing and Balance Center who worked as facilitator during the concert and provided useful suggestions for the design of the experiments connected to the concert.

10. REFERENCES

[1] T. Lenarz, "Cochlear implant–state of the art," *Laryngo-rhino-otologie*, vol. 96, no. S 01, pp. S123–S151, 2017.

[2] E. Ambert-Dahan, A.-L. Giraud, O. Sterkers, and S. Samson, "Judgment of musical emotions after cochlear implantation in adults with progressive deafness," *Frontiers in Psychology*, vol. 6, p. 181, 2015.

[3] M. V. Certo, G. D. Kohlberg, D. A. Chari, D. M. Mancuso, and A. K. Lalwani, "Reverberation time influences musical enjoyment with cochlear implants," *Otology & neurotology*, vol. 36, no. 2, pp. e46–e50, 2015.

[4] K. Gfeller, A. Christ, J. F. Knutson, S. Witt, K. T. Murray, and R. S. Tyler, "Musical backgrounds, listening habits, and aesthetic enjoyment of adult cochlear implant recipients," *Journal of the American Academy of Audiology*, vol. 11, no. 07, pp. 390–406, 2000.

[5] A. Nagathil, J.-W. Schlattmann, K. Neumann, and R. Martin, "Music complexity prediction for cochlear implant listeners based on a feature-based linear regression model," *The Journal of the Acoustical Society of America*, vol. 144, no. 1, pp. 1–10, 2018.

[6] G. Dritsakis, R. M. van Besouw, and A. O'Meara, "Impact of music on the quality of life of cochlear implant users: a focus group study," *Cochlear Implants International*, vol. 18, no. 4, pp. 207–215, 2017.

[7] W. Nogueira, A. Nagathil, and R. Martin, "Making music more accessible for cochlear implant listeners: Recent developments," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 115–127, 2018.

[8] E. Schubert, J. Marozeau, C. J. Stevens, and H. Innes-Brown, "'like pots and pans falling down the stairs'. experience of music composed for listeners with cochlear implants in a live concert setting," *Journal of New Music Research*, vol. 43, no. 2, pp. 237–249, 2014.

[9] N. Zangwill, "Listening to music together," *aesthj Journal*, vol. 52, no. 4, pp. 379–389, 2012.

[10] R. Y. Litovsky, A. Parkinson, and J. Arcaroli, "Spatial hearing and speech intelligibility in bilateral cochlear implant users," *Ear and hearing*, vol. 30, no. 4, p. 419, 2009.

[11] A. Habibi, G. Kreutz, F. Russo, and M. Tervaniemi, "Music-based interventions in community settings: Navigating the tension between rigor and ecological validity," *Annals of the New York Academy of Sciences*, vol. 1518, no. 1, pp. 47–57, 2022.

[12] M. Tervaniemi, "The neuroscience of music–towards ecological validity," *Trends in Neurosciences*, 2023.

[13] S. McAdams, *Music Perception*, vol. 22, no. 2, p. 171, 2004.

[14] S. Thompson, "Audience responses to a live orchestral concert," *Musicae Scientiae*, vol. 10, no. 2, pp. 215–244, 2006.

[15] M. C. Broughton, E. Schubert, D. G. Harvey, and C. J. Stevens, "Continuous self-report engagement responses to the live performance of an atonal, post-serialist solo marimba work," *Psychology of Music*, vol. 47, no. 1, pp. 109–131, 2019.

[16] A. Czepiel, L. K. Fink, L. T. Fink, M. Wald-Fuhrmann, M. Tröndle, and J. Merrill, "Synchrony in the periphery: inter-subject correlation of physiological responses during live music concerts," *Scientific reports*, vol. 11, no. 1, p. 22457, 2021.

[17] D. Swarbrick, F. Upham, C. Erdem, A. R. Jensenius, and J. K. Vuoskoski, "Measuring virtual audiences with the musiclab app: Proof of concept," in *Proceedings of the SMC Conferences*. SMC Network, 2022.

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

[19] E. Schubert, "Using affect valence and emotion valence to understand musical experience and response: the case of hated music," in *Proceedings of the 30th ISME World Conference on Music Education Music Pædeia: From ancient Greek philosophers toward global music communities*. International Society for Music Education Nedlands, WA, 2012, pp. 317–324.

[20] C. J. Limb and J. T. Rubinstein, "Current research on music perception in cochlear implant users," *Otolaryngologic Clinics of North America*, vol. 45, no. 1, pp. 129–140, 2012.

[21] M. Wald-Fuhrmann, H. Egermann, A. Czepiel, K. O'Neill, C. Weining, D. Meier, W. Tschacher, F. Uhde, J. Toelle, and M. Tröndle, "Music listening in classical concerts: Theory, literature review, and research program," *Frontiers in psychology*, vol. 12, p. 638783, 2021.

[22] T. Turino, *Music as social life: The politics of participation*. University of Chicago Press, 2008.

[23] M. Slater, "Immersion and the illusion of presence in virtual reality," *British journal of psychology*, vol. 109, no. 3, pp. 431–433, 2018.

[24] J. K. O'Regan and A. Noë, "What it is like to see: A sensorimotor theory of perceptual experience," *Synthese*, vol. 129, pp. 79–103, 2001.

[25] H. Lee, "A conceptual model of immersive experience in extended reality," 2020.

[26] S. Agrewal, A. M. D. Simon, S. Bech, K. B. Bærentsen, and S. Forchammer, "Defining immersion:: literature review and implications for research on audiovisual experiences," *Journal of the Audio Engineering Society*, vol. 68, no. 6, pp. 404–417, 2020.

[27] S. Agrawal, S. Bech, K. De Moor, and S. Forchhammer, "Influence of changes in audio spatialization on immersion in audiovisual experiences," *Journal of the Audio Engineering Society*, vol. 70, no. 10, pp. 810–823, 2022.

[28] J. Williams, J. Francombe, and D. Murphy, "Exploring the influence of multichannel soundtracks on film immersion," in *Audio Engineering Society Conference: AES 2023 International Conference on Spatial and Immersive Audio*. Audio Engineering Society, 2023.

[29] J. Francombe, T. Brookes, and R. Mason, "Evaluation of spatial audio reproduction methods (part 1): elicitation of perceptual differences," *Journal of the Audio Engineering Society*, vol. 65, no. 3, pp. 198–211, 2017.

[30] F. Rumsey, "Immersive audio—defining and evaluating the experience," *Journal of the Audio Engineering Society*, vol. 68, no. 5, pp. 388–392, 2020.

[31] Y. Wycisk, K. Sander, R. Kopiez, F. Platz, S. Preihs, and J. Peissig, "Wrapped into sound: Development of the immersive music experience inventory (imei)," *Frontiers in Psychology*, vol. 13, p. 951161, 2022.

[32] H. Innes-Brown, A. Au, C. Stevens, E. Schubert, and J. Marozeau, "New music for the bionic ear: An assessment of the enjoyment of six new works composed for cochlear implant recipients," in *12th International Conference on Music Perception and Cognition (ICMPC) 8th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*, 2012.

[33] J. J. Galvin III, Q.-J. Fu, and G. Nogaki, "Melodic contour identification by cochlear implant listeners," *Ear and hearing*, vol. 28, no. 3, p. 302, 2007.

[34] A. R. Bargum, D. Kandpal, O. I. Kristjansson, S. R. Mosen, J. Andersen, and S. Serafin, "Virtual reconstruction of a the ambisonic concert hall of the royal danish academy of music," in *2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2021, pp. 99–102.

[35] A. R. Herzog and J. G. Bachman, "Effects of questionnaire length on response quality," *Public opinion quarterly*, vol. 45, no. 4, pp. 549–559, 1981.

[36] J. A. Holman, A. Drummond, S. E. Hughes, and G. Naylor, "Hearing impairment and daily-life fatigue: A qualitative study," *International Journal of Audiology*, vol. 58, no. 7, pp. 408–416, 2019.

[37] R. G. Burgess, *In the field: An introduction to field research*. Routledge, 2002, vol. 8.

[38] S. Brinkmann, "Unstructured and semi-structured interviewing," *The Oxford handbook of qualitative research*, vol. 2, pp. 277–299, 2014.

[39] K. E. Gfeller, C. Olszewski, C. Turner, B. Gantz, and J. Oleson, "Music perception with cochlear implants and residual hearing," *Audiology and Neurotology*, vol. 11, no. Suppl. 1, pp. 12–15, 2006.

[40] R. Y. Litovsky, H. S. Colburn, W. A. Yost, and S. J. Guzman, "The precedence effect," *The Journal of the Acoustical Society of America*, vol. 106, no. 4, pp. 1633–1654, 1999.

[41] A. Au, J. Marozeau, H. Innes-Brown *et al.*, "Music for the cochlear implant: audience response to six commissioned compositions," in *Seminars in Hearing*, vol. 33, no. 04. Thieme Medical Publishers, 2012, pp. 335–345.

THE ELUSIVE NATURE OF INATTENTIONAL DEAFNESS: ASSESSING THE INFLUENCE OF VISUAL ATTENTION ON BACKGROUND MUSIC PERCEPTION IN TV PROGRAMMES

Roser BATLLE-ROCA (roser.batlle@upf.edu) [1], **Diana TYMAN**[1], **Blai MELÉNDEZ-CATALÁN**[2], **Emilio MOLINA**[2], **Xavier SERRA**[1], and **Perfecto HERRERA-BOYER**[1]

[1]*Music Technology Group*, **Universitat Pompeu Fabra**, Barcelona, Spain
[2]**BMAT Licensing, S. L.**, Barcelona, Spain

ABSTRACT

Evaluating the audibility level of background music in TV programmes is a significant challenge as some music industry copyright regulations arrange music remuneration accordingly. Besides the loudness level, other characteristics, such as sensory attention, may influence its perception, raising or lowering its audibility threshold. Yet, there is limited literature exploring this particular problem. Our study examines how visually motivated attention impacts the perception of background music in TV programmes, contributing to *inattentional deafness*—the failure to perceive auditory stimuli due to visual perceptual load. We conducted two experiments based on forced-choice and detection tasks focused on assessing the influence of visually motivated attention on background music perception in TV programmes. Experiment 1 shows that participants may experience inattentional deafness when visually motivated, but not strongly enough to support our hypothesis. Hence, we refined our methodology in Experiment 2 with a dual-task paradigm to guarantee forced visual-motivated attention. Analysis via a one-way ANOVA demonstrates a statistically significant influence of forced visual attention (Task 1) towards the music perception assignment (Task 2). Thus, our findings indicate that the audibility of background music is subject to the visual stimuli load of the TV programme.

1. INTRODUCTION

Music in audiovisual productions represents a substantial source of income for the music industry through copyright retribution. Particularly within TV programmes, multiple types of content include music, whether as a primary auditory element or background accompaniment. Royalty distribution regulations vary across countries and consider several conditions, such as broadcasting time slots and the importance of music in the production.

In 2017, the WIPO (World Intellectual Property Organization) arbitrated that distribution rights of background music in TV programmes should be subject to a 3-level audibility scale [1]: (1) non-audible or inaudible, (2) *barely audible*, and (3) audible. Full distribution rights are granted only when the music is audible. When the music is *barely audible*, 50% of the stipulated royalties are effectively paid, and no royalties are paid when the music is not audible. However, there is no standardised method to evaluate and classify the background music audibility level in the context of TV programmes. Still, collective management organisations, such as SGAE (General Society of Authors and Publishers) in Spain, must estimate this component. This issue is further complicated due to the structure of the broadcasted signal, which is a mixture of music and other sounds (e.g. speech, sirens, applause) emitted by TV broadcasters, together with the visual component of the TV programme (Figure 1 [1]). Thus, the received signal is composed of the visual and auditory components, the latter including both musical and non-musical sources simultaneously.

Despite evaluating background music audibility levels has become a crucial research problem for this particular context, little literature explores this issue [2, 3]. Both identified publications primarily focus on defining an audibility threshold level for the class *barely audible*. Besides music loudness level, many characteristics may influence background music perception, including sensory attention, listening conditions, environmental noise, and masking by other sounds within the mixture such as applause or sirens. All these elements might contribute to raising or lowering music's audibility threshold.

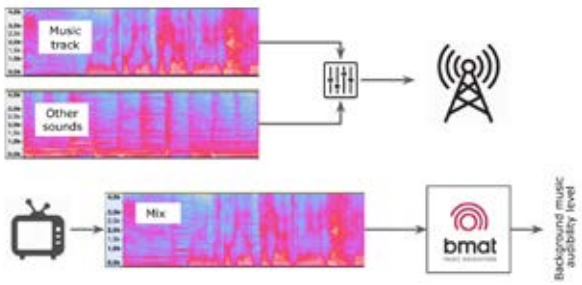


Figure 1. Representation of the broadcasted audio signal.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] BMAT is one of the principal companies specialised in monitoring and reporting music usage across various platforms, including radio, television, and online streaming services. More information at: https://www.bmat.com/.

Furthermore, inattentional deafness, defined as the failure to perceive auditory stimuli due to visual perceptual load [4], emerges as an intriguing concept in this discussion, adding complexity to the understanding of how background music is perceived in audiovisual productions. Koreimann et al. [5] support the existence of inattentional deafness in music, highlighting its potential impact on music perception within audiovisual environments.

In this study, we aim to explore the effects of inattentional deafness on the perception of background music in the context of TV programmes. We introduce two experiments based on force-choice and detection tasks [6], reviewed and approved by the Institutional Committee for Ethical Review of Projects (CIREP) at Universitat Pompeu Fabra [2]. In both experiments, we use real broadcasted TV content and implement a condition to motivate visual attention, trying to induce inattentional deafness in the subjects. Despite sharing the hypothesis and research purpose, the methodology for inducing inattentional deafness differs between the two experiments due to the insufficiency of Experiment 1 in supporting our hypothesis. Thus, we refined our approach in Experiment 2 by adopting a dual-task paradigm to guarantee forced visually motivated attention. Detailed descriptions of the methodology followed in each experiment are provided in Section 3.1 and 4.1, respectively. Our findings indicate that the audibility of background music is subject to the visual stimuli load of the TV programme, which is significant to the estimation of the audibility level of background music. Therefore, this characteristic should be taken into account when assessing the audibility level for copyright retribution, when following the 3-level audibility scale.

2. BACKGROUND AND RELATED WORK

2.1 Inattentional deafness

The perception of the loudness of auditory signals is naturally subjective. Each subject's perception can vary due to individual features such as attention and awareness. Attention involves directing the mind towards processing information aligned with the goals of the current task [7], while awareness refers to acknowledging that an event, an object or a sensory pattern is happening or exists [8]. Auditory scene analysis enables the human auditory system to discern and perceive sound events in the environment as distinct auditory objects [9]. For instance, when talking with someone in the street, where there are external noises, we can focus our attention on the voice and isolate the street noises, i.e. car engines and other people talking. The voice, hence, becomes the foreground, whereas the rest of the sonic input is treated as background, receiving much less attention. However, multiple factors can influence the perception of a sound within a TV programme, from masking by another auditory stimuli (e.g. applause or voice) to external environment noise and sensory attention, among others.

Focusing on sensory attention, the concept of inattentional deafness, defined by Macdonald and Lavie [4], occurs when individuals fail to perceive auditory stimuli because of the attention given to the visual stimuli. Multiple investigations demonstrate the influence of visual stimuli on inattentional deafness through different experiments [4, 10–12]. As exposed by Koreimann et al. [5], inattentional deafness in music is defined as the inability to consciously perceive characteristics of the music stimuli when the attention is motivated towards a particular element of the piece (i.e. timpani beats). Their investigation sustains the existence of inattentional deafness in music and that the human auditory system is more influenced by attentional distraction than previously thought. In addition, they point out the potential impact of inattentional deafness in music perception within the audiovisual domain.

2.2 Related Work

Estimating background music audibility levels within TV programmes considering the arbitration by WIPO remains relatively underexplored in the literature. There are two significantly related works targeting this particular research scope.

On the one side, López and Ramallo [2] propose that the feasible solution is to build a deterministic system that converges to the average estimate of listeners' audibility, despite considering several elements that may influence the perception of background music, such as environment noise, musical sensitivity of the listeners and listening environment. Their study consists of a manual annotation of 88 artificial 10-second clips of video composed of the voice of two popular Spanish news hosts (one male, one female) and added, in the background, songs of different music genres. Participants must adjust the loudness level to be marginally audible. Consequently, the authors establish the *barely audible* threshold around -30 dB LUFS and propose a linear regression to adjust the perceived loudness level. Nevertheless, they conclude that, when working with real broadcasted content, where voice and music are mixed, it is required to use source separation techniques to estimate the music loudness level. However, our own examinations suggest that these techniques are still far from providing excellent results that can be trusted in such a sensitive application scenario.

On the other side, our previous investigation in this topic [3] explored the blurry limit between audibility and inaudibility, intending to define *barely audible*. We carried out two experiments based on broadcasted TV content to assess the audibility threshold levels of the class *barely audible* and understand the effects of the listening conditions. We estimated the sound-to-music ratio for *barely audible* to be between -22.7 dB LUFS and -36.4 dB LUFS when watching the content from a television. Our findings suggested that background music perception is affected by loudness level, listening condition, music sensitivity and type of TV content. Thus, we believe there is considerable

[2] The proposed experiments approach was reviewed and approved by the Institutional Committee for Ethical Review of Projects stating compliance with the ethical principles in research involving humans and personal data protection regulations (guided by Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation)).

scope for future research to consider multiple characteristics that might influence background music perception.

With the present study, we aim to complement our previous work by exploring the effects of inattentional deafness on background music perception. We acknowledge the potential significance of this phenomenon within our research scope.

3. EXPERIMENT 1

3.1 Methodology

In Experiment 1, we aim to investigate how directing attention to visual stimuli affects the perception of background music. We hypothesise that forcing attention to the visual stimuli will make background music less noticeable due to inattentional deafness.

We designed an experiment with a between-subjects approach based on two conditions: *active*, where the participants were explicitly instructed to focus their attention on the visual part of the video, and *passive*, where participants were inclined to view the clip without any specific instruction to direct their attention. Participants were randomly assigned to one of the two conditions.

The experiment involved a force-choice task and used 68 10-second video clips of real TV programmes. After viewing each clip, participants were asked to answer a yes-no question about the content, referring to the image, speech or music within. Questions were carefully balanced on yes-no responses and considered both the type of question and direction of sensory attention (visual or auditory) to minimize bias and avoid guiding participants towards our research goal. Specifically, questions were divided into 50% related to visual modality and 50% to auditory modality. The latter category was further divided into 41.2% for speech and 58.8% for music (representing 20.6% to speech and 29.4% to music of the total questions). Experiment 1 was conducted online and was available in English and Spanish.

3.1.1 Material and Tools

For this experiment, we used 10-second clips of video recorded between 28th June 2021 at 13:15 and 7th July 2021 at 10:15. Spanish stimuli were retrieved from five Spanish TV channels: *TV1*, *TV2*, *LaSexta*, *Cuatro* and *Antena 3*. Instead, English stimuli were obtained from three TV channels in the UK: *BBC*, *Channel 5* and *CBBC*; and three from the US: *CBS*, *NBC* and *Fox*. These stimuli were extracted from BMAT's database and cannot be openly shared as they are subject to confidentiality regulations. The experiment was built with lab.js [3] and distributed through the Pavlovia platform [4].

3.1.2 Participants

A total of 85 subjects participated in the experiment (EN=43, ES=42). One English-version participant was excluded from the study because of a greater number of errors than any other participant. An equal number of valid participants in both language groups was then achieved. Both attention categories (active and passive) had an equal distribution of participants, with 21 participants in each condition per language. A two-way ANOVA supports there is no statistically significant difference between languages (F(1, 1)=2.381, p=0.127). Hence, results are analysed together. Moreover, 30 participants completed the experiment through Prolific [5] platform and were compensated with 5€. The rest of the participants were volunteers. Two independent one-way ANOVA on the number of errors (F(1, 82)=0.003, p=0.960) and the reaction time (F(1, 82)=0.62, p=0.805) demonstrate that there is no statistically significant difference between the behaviour of the participants depending on their origin. Therefore, all retrieved data are analysed together.

Regarding headphone usage, participants predominantly used in-ear or circumaural headphones (77.9%). A smaller representation used supraural headphones (4.3%), while only 7.1% used noise-cancelling headphones. Most participants reported their environmental noise as *weak* or *moderate*, with only 3.6% experiencing *strong* noise, none reporting *very noisy* conditions, and 9.5% perceiving their environment as *inaudible*. On average, the experiment duration was 25 minutes.

3.2 Results

Participants answered on average 75.4% of the forced-choice questions correctly. However, according to the question type, the average amount of errors is diverse: 19.8% for image questions, 17.4% for speech questions and 37.6% for music questions. Notably, the error rate for the music-specific questions is higher than the other two categories. When considering attention groups (active or passive), performance is similar in both cases; including speech and music-based questions, which we hypothesised would show a worse performance in the active experimental condition. Indeed, no statistically significant difference is observed when performing a one-way ANOVA between the two attention groups (F(1, 5710)=0.094, p=0.759). Hence, our hypothesis is not supported by these results. Despite this situation, our results show that the reaction time in the active group was faster than for the passive group (active=2872.1ms, passive=3363.3ms) and this difference is statistically significant (F(1, 5642)=16.596, p=4.7×10^{-5}).

At the end of the experiment, we included a subjective question to know how much background music the participants perceived. Even though six out of the 68 presented clips did not have background music (8.8%), for most participants 60% of the clips had background music. Hence, around 30% of background music tends to be ignored or not acknowledged by the participants.

We suspect that our approach to induce inattentional deafness in the active group was not powerful enough and, to truly focus attention on non-auditory stimuli stronger manipulations than an instruction in the beginning of the experiment should be used. Consequently, we proposed a

[3] https://lab.js.org/
[4] https://pavlovia.org/
[5] https://www.prolific.com/

new and refined experiment to observe the influence of visually motivated attention on the perception of background music in TV programmes: Experiment 2.

4. EXPERIMENT 2

4.1 Methodology

In the present investigation, Experiment 2 intends to explore the impact of visually motivated attention on the perception of background music in the context of TV programmes, with a different approach than Experiment 1. We hypothesise that focusing participants' attention towards the visual component may diminish music perception due to inattentional deafness.

Experiment 2 was based on a dual-task paradigm with two consecutive detection tasks. The first task (Task 1) involved an attention distraction, where the participants were instructed to press the spacebar once they detected a specific visual element within a clip of video, such as a yellow shirt or a dog. With this task, we aimed to force and guarantee continued visually motivated attention to induce inattentional deafness. Then, the second task (Task 2) consisted of an attention and perception evaluation, where participants answered to three true-or-false statements each related to a different characteristic of the video: image, speech and music. All statements were balanced in true and false responses. This approach ensured that participants remained unaware of the experiment's primary goal: assessing background music perception.

This experiment comprised 65 10-second video extracts from Spanish TV, with 55 including background music and 10 without. In addition, at the beginning of the experiment, three extra clips were presented to familiarize participants with the task. To maintain consistent auditory conditions within subjects, the experiment was carried out on-site and exclusively in Spanish.

4.1.1 Material and Tools

Each experiment session took place in a designated room with an average ambience noise of 35.8 dBA. The room was displayed to recreate a living room ambience with a TV and a sofa from which the participants completed the experiment. On average, participants' ears were positioned approximately 2.06 meters away from the TV and the volume was set approximately at 58 dBA. The experiment was directly supervised by one of the responsible researchers.

For this experiment, we also used 10-second segments from the TV programmes recorded between 28th June 2021 at 13:15 and 7th July 2021 at 10:15. The clips of video were selected from the same five Spanish TV channels as for Experiment 1. However, we chose different clips for Experiment 2 to not use the same audiovisual stimuli as in Experiment 1. We used the same tools as in Experiment 1 to design, implement and distribute the experiment.

4.1.2 Participants

Experiment 2 was completed by 27 participants, all of whom were volunteers and native Spanish speakers. However, two of them had to be discarded due to technical issues during the experiment that disrupted the session recording. Another participant was excluded from the analysis due to poor performance, particularly in Task 1, where their accuracy was below 50%. As their performance did not meet the minimum threshold, we cannot guarantee that they maintained the intended focus on the visual content. Thus, we consider the results of 24 valid participants.

In addition, some technical disruptions prompted a few participants to encounter minor issues during their sessions. Specifically, nine subjects did not visualise one clip, and one subject did not visualise two clips (is, 11 missed evaluations). However, all non-visualised clips were different. While these technical issues occurred, we believe that the impact on our overall interpretation of the results is minimal. This issue has been taken into account and participants' average performance has been adjusted accordingly.

4.2 Results

Experiment 2 employed a dual-task paradigm intending to direct participants' attention to the visual component of the audiovisual stimuli while observing the impacts of this focus on music perception. In Task 1, participants were instructed to detect a specific element on the video and press the space bar when it appeared. On average, participants completed Task 1 with 75.6% accuracy, indicating their focus on the visual component while watching the clips.

Task 2 involved three true-or-false statements referring to the image, speech and music components of the clips. The error rate varied depending on the type of assignment-task, with a 25% error rate for image statements, 25.5% for speech and 35.5% for music. Notably, statements related to music-specific characteristics exhibited a 10-point higher error rate compared to the other two categories. On average, participants required approximately 8 seconds to decide whether the presented statements were true or false (M=7.75, SD=3.12).

To observe the influence of Task 1 over Task 2, we conducted three one-way ANOVA considering the errors of both tasks. That is, Task 1 errors as independent variable and each Task 2 characteristic-specific errors as dependent variables. In Table 1, we depicted the results of this analysis. With this statistical analysis, we detected statistically significant differences on the errors for both auditory related-tasks, (speech: $F(1, 1550)=3.976$, $p=0.046$; music: $F(1, 1550)=1.700$, $p=0.006$), due to the differential attention load induced in the visual and auditory domains.

In addition, we analysed the effects of completing Task 1, whether correct or incorrect, on Task 2. Specifically, we examined the error rates of Task 2 depending on Task 1 performance. Considering the results reported in Table 2, we noticed that image-specific questions exhibit the same error rates regardless of Task 1 performance. Instead, both auditory domain-related statements showed improved accuracy when Task 1 was incorrect. Specifically, for speech-specific statements, performance increased by 5 percentage points, and for music-specific statements, by 7 per-

Table 1. One-way ANOVA demonstrating the effects of Task 1 on the errors in each of the different characteristic-specific categories in Task 2. A statistically significant difference is considered when $p < 0.05$ (in bold).

		Sum of squares	Df	Mean Square	F	p
T2-Image	Between groups	0.005	1	0.005	0.029	0.865
	Within groups	290.995	1550	0.188		
	Total	291.000	1551			
T2-Speech	Between groups	0.755	1	0.755	3.976	**0.046**
	Within groups	294.204	1550	0.190		
	Total	294.959	1551			
T2-Music	Between groups	1.700	1	1.700	7.462	**0.006**
	Within groups	353.098	1550	0.228		
	Total	354.798	1551			

Table 2. Task 2 characteristic-specific error rates concerning Task 1 error rates (where 0 indicates correct responses and 1 indicates errors). The *Mean* row indicates the average error rates per each category within Task 2.

T1	T2-Image	T2-Speech	T2-Music
0	0.25	0.27	0.37
1	0.25	0.22	0.3
Mean	0.25	0.26	0.35

centage points. This is aligned with the results reported in Table 1, where a statistically significant difference is demonstrated for the auditory domain assignments.

Similarly to Experiment 1, at the end of the experiment, participants were asked to indicate the percentage of videos they thought contained background music. For most participants, the median perceived percentage of videos containing background music was 65%. However, it is noteworthy that approximately 85% of the videos contained background music. This discrepancy highlights a tendency for participants to underestimate and ignore background music in TV programmes.

5. DISCUSSION

In the context of audiovisual productions, specifically for broadcasted TV programmes, we aimed to study the influence of visually motivated attention on the perception of background music audibility due to inattentional deafness. For this particular case study, we presented two experiments intending to explore the effects of inattentional deafness on music perception.

Experiment 1 wanted to observe the presence of inattentional deafness by assessing the differences between a group actively motivated to pay attention to image contents of a video, and a passive group, which did not receive any specific instruction. However, results between the two groups do not present statistically significant differences, and participants' performance is similar independently of the assigned group. Despite not demonstrating any significant effect of attentional motivation on background music perception, we observe an influence of attentional motivation on reaction time depending on the attention group. Consequently, we consider that our methods could be improved and we propose a redefined experiment.

We modified our methodology in four key aspects. Firstly, we introduced a dual-task paradigm approach based on two consecutive tasks: (i) visual attention distraction and (ii) perception evaluation. This approach aimed to enhance participants' focus on the visual stimuli of the presented clip, inducing more forced visual attention rather than trusting a basic instruction as in Experiment 1. Secondly, we designed Task 2 to include three statements related to the image, speech, and music characteristics of the clip, providing more insights into music perception compared to the single-question format used in Experiment 1. In Experiment 1, only 29.4% of the clips included a music-specific question. Instead, in Experiment 2, we have an evaluation of music perception per every displayed clip. Thirdly, we conducted the experiment onsite to ensure a homogeneous environment across participants and to have more control over the experiment setting conditions. Finally, we focused our study on exclusively Spanish TV content. Since this investigation takes into account the 2017 arbitration by WIPO affecting SGAE and, it is developed within the scope of Spanish funding, we decided to prioritize Spanish TV programmes. We limited participants to be Spanish native speakers living in Spain. With this restriction, we wanted to reduce any variability due to different divergences in popular TV content. These modifications were made to correct the experimental design of Experiment 1.

Findings in Experiment 2 demonstrate the presence of inattentional deafness as the two auditory domain-related queries in Task 2 are detrimentally affected by the attentional focus required by Task 1. Not only does this indicate that Task 1 fulfilled the goal of inducing inattentional deafness in the participants, but it also shows the relevance of this phenomenon on auditory perception. For background music perception, a one-way ANOVA analysis shows a statistically significant difference in the music perception assignment (Task 2) due to the directed attention to the visual component (Task 1). Moreover, our results indicate that performance on auditory-related statements in Task 2 is subject to the accuracy of Task 1. That is, when Task 1 was error-prone, participants obtained better results for speech and music statements in Task 2, whereas performance remained stable for the image-related statements.

Considering all the modifications to our methodology approach and the results obtained, we can be certain that those were effective. We were able to validate our hypoth-

esis and demonstrate the effects of inattentional deafness in the context of broadcasted TV. These findings suggest that background music audibility within a TV programme context is subject to inattentional deafness. In addition, our findings expand the range of scenarios where inattentional deafness has been reported, contributing to its consolidation as a perceptual phenomenon deserving more research attention. These results question the structure of copyright distribution regarding the three audibility level categories established by WIPO.

The research reported here indicates how multidimensional and complex auditory perception in an audiovisual context can be. Regulations and arbitrations regarding music rights distribution should be very cautious in proposing artificial constructs that, when measured at the reception place (or its proxy) cannot be determined with the right precision. In this sense, our research prompts for promoting and using metadata at the source, when direct methods of estimation can be easily obtained. In our case, TV stations could easily provide a speech-to-music ratio that, even not being the only factor to be considered, would provide a solid ground to decide on the *barely audible* case. Nonetheless, we believe that contextual variables influencing background music perception should be taken into account when assessing background music audibility within audiovisual productions, specifically for TV programmes.

6. CONCLUSION

We present a study focused on assessing the effects of inattentional deafness of background music perception in the context of TV programmes. Evaluating the audibility level of background music is significant within the music industry, as some copyright regulations set music remunerations accordingly. Literature demonstrates that many characteristics may influence music perception, beyond its loudness level. We expand previous investigations by focusing on inattentional deafness in background music perception. We proposed two experiments based on force-choice and detection tasks. Our first experiment did not demonstrate strongly enough the phenomenon of inattentional deafness of background music. Therefore, we redefined our methodology to guarantee forced visually motivated attention to induce inattentional deafness. Our results demonstrate the presence of inattentional deafness, as the auditory-domain assignments present statistically significant differences concerning the induced attention to the visual component of the clip. In addition, our findings show statistical significance when perceiving background music in TV programmes.

Thus, the present investigation contributes to untangling the assessment of background music in TV programmes by determining inattentional deafness due to visual stimuli a significant characteristic to take into account. We believe that copyright regulations considering rights distributions based on the three auditory levels proposed by WIPO (audible, *barely audible* and inaudible) should consider other phenomena and circumstances that influence perception in addition to the loudness level, such as sensory attention and inattentional deafness.

Acknowledgments

This research was completed under task 1.3 of the "AI system for automatic audibility estimation of background music in audiovisual productions" project, known as the LoudSense project, funded by ACCIÓ INNOTEC-2020, with grant number ACE014/20/000051. This work is partially supported by *IA y Música: Cátedra en Inteligencia Artificial y Música* (TSI-100929-2023-1), funded by the *Secretaría de Estado de Digitalización e Inteligencia Artificial*, and the European Union-Next Generation EU, under the program *Cátedras ENIA 2022 para la creación de cátedras universidad-empresa en IA*.

7. REFERENCES

[1] SGAE, "SGAE statement on the WIPO arbitration award ending the conflict of night time slots on television," 2017, (Last Accessed 21/08/2021). [Online]. Available: http://www.sgae.es/en-EN/SitePages/EstaPasandoDetalleActualidad.aspx?i=2190&s=5

[2] J. J. López and S. Ramallo, "Study of the audibility of background music in TV programs: Towards a normative proposal," in *Audio Engineering Society Convention 152*. Audio Engineering Society, 2022.

[3] R. Batlle-Roca, P. Herrera-Boyer, B. Meléndez-Catalán, E. Molina, and X. Serra, "Towards a characterization of background music audibility in broadcasted TV," *International Journal of Environmental Research and Public Health*, vol. 20, no. 1, 2023. [Online]. Available: https://www.mdpi.com/1660-4601/20/1/123

[4] J. S. Macdonald and N. Lavie, "Visual perceptual load induces inattentional deafness," *Attention, Perception, and Psychophysics*, vol. 73, pp. 1780–1789, 2011.

[5] S. Koreimann, B. Gula, and O. Vitouch, "Inattentional deafness in music," *Psychological Research*, vol. 78, pp. 304–312, 2014.

[6] V. Pulkki and M. Karjalainen, *Communication Acoustics An Introduction to Speech, Audio and Psychoacoustics*. United Kingdom: John Wiley Sons, 2015.

[7] K. Nobre and S. Kastner, *The Oxford handbook of attention*. Oxford Library of Psychology, 2014.

[8] K. Gafoor, "Considerations in measurement of awareness national seminar on emerging trends in education,(november), 1–6," 2012.

[9] E. S. Sussman, "Auditory scene analysis: An attention perspective," *Journal of Speech, Language, and Hearing Research*, vol. 60, pp. 2989–3000, 2017.

[10] N. Lavie, "Distracted and confused?: Selective attention under load," *Trends in Cognitive Sciences*, vol. 9, pp. 75–82, 2005.

[11] D. Raveh and N. Lavie, "Load-induced inattentional deafness," *Attention, Perception, and Psychophysics*, vol. 77, pp. 483–492, 2015.

[12] K. Molloy, T. D. Griffiths, M. Chait, and N. Lavie, "Inattentional deafness: Visual load leads to time-specific suppression of auditory evoked responses," *Journal of Neuroscience*, vol. 35, pp. 16 046–16 054, 2015.

STATISTICAL ANALYSIS OF MUSICAL FEATURES FOR EMOTIONAL SEMANTIC DIFFERENTIATION IN HUMAN AND AI DATABASES

Francisco BRAGA (0000-0002-9094-6320) [1*], **Jorge FORERO** (0000-0002-4662-8325) [2*], and **Gilberto BERNARDES** (0000-0003-3884-2687) [2]

[1] **NOVA University of Lisbon, School of Science and Technology & NOVA LINCS**, Lisbon, Portugal
[2] **University of Porto, Faculty of Engineering & INESC TEC**, Porto, Portugal
[*] These authors contributed equally.

ABSTRACT

Understanding the structural features of perceived musical emotions is crucial for various applications, including content generation and mood-driven playlists. This study performs a comparative statistical analysis to examine the association of a set of musical features with emotions, described using adjectives. The analysis uses two datasets containing rock and pop musical fragments, categorized as human-generated and AI-generated.

Focusing on four emotional adjectives (happy, sad, angry, tender-gentle) representing each valence-arousal plane's quadrant, we analyzed semantic differential meanings reported as symmetric pairs for all possible combinations of quadrants through diagonals, vertical, and horizontal axes.

The results obtained were discussed based on Livingstone's circular representation of emotional features in music.

Our findings demonstrate that the human and AI-generated datasets could be considered equivalent for diagonal symmetries, while horizontal and vertical symmetries show discrepancies. Furthermore, we assessed significant separability for both happy-sad and angry-tender pairs in the human dataset. In contrast, the AI-generated music exhibits a strong differentiation mainly in the angry-gentle pair.

1. INTRODUCTION

Music has long been associated with emotions and systematically studied across various disciplines since the beginning of the XX century.

The emotion felt and perceived in a musical fragment has been a topic of research since then, mainly using conceptual frameworks taken from psychology, such as categorical [1] and dimensional [2] models.

Diverse tests have been proposed to measure emotions from experiential, physiological, or behavioral perspectives [3]. Self-reports are commonly used for experimental evaluation. When using self-reports for categorical classification, participants label audio using adjectives. In contrast, a semantic differential approach is generally used in self-reports for dimensional analysis, where participants rank a pair of differential adjectives to build an emotional representation space, such as the valence-arousal plane. [1] Thus, language, or how we appraise emotions through these adjectives, is at the root of the experiential measurement methodology [5].

Furthermore, automatic music generation conditioned on textual description or melodic features has significantly progressed in recent years. In particular, some models, such as MusicLM [6] and MusicGen [7], have produced musical fragments within a short period of consistency, generated by prompting text with high-level music descriptions.

A crucial step in investigating the role of emotion in music involves identifying acoustic and structural elements reliably linked to emotional connotations.

Livingstone's circular representation of music and emotion [8] synthesizes an accumulation of structural features derived from studies reviewed by Schubert [9], Gabrielsson and Lindstrom [10], and Gabrielsson and Juslin [11] (see Figure 1). In this visualization, the placement of features along each axis of the valence-arousal plane corresponds to the frequency of studies indicating the association, with features positioned in quadrants if the association is confirmed across multiple independent studies.

To the best of our knowledge, no study has yet provided an empirical and systematic approach to statistical emotion differentiation using musical features within human-made music and semantically conditioned AI-generated music.

Although other research focuses primarily on human-generated datasets, exploring the relationship between musical features and perceived emotions [12], there is no evidence of a methodology incorporating statistical analysis to empirically compare and contrast AI-generated music.

This study aims to analyze emotion-related features in a dataset of music generated by a deep learning model and compare it with an existing human-made and annotated music-emotion dataset. We seek to investigate the confidence level over a set of musical features that differentiate emotional adjective pairs along the valence-arousal plane in both datasets. The comparison aims to highlight the differences and similarities in emotional expression between human and AI-generated music, contributing to the

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] The adoption of semantically differential or polarized adjectives to infer meaning originated from Osgood's work [4].

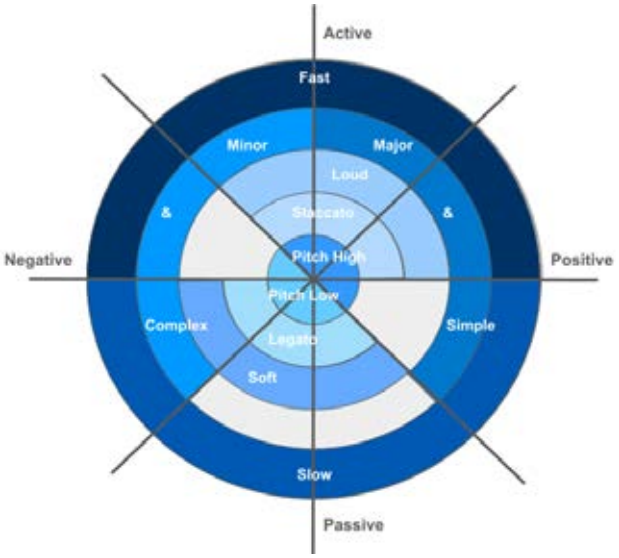


Figure 1. Livingstone Meta-analysis of emotional cues in music.

Figure 2. Emotional Adjectives and Symmetries in the Valence/Arousal space.

broader understanding of music's emotional conveyance for both scenarios.

The remainder of this paper is structured as follows. Section 2 describes the research methodology used, exposing the audio database used and the musical features selection. Section 3 provides the Results, proposing a radial plot to analyze the data. Finally, Section 4 offers a discussion and concludes in Section 5.

2. RESEARCH METHODOLOGY

We compared six pairs of adjectives distributed in the valence arousal plane and reported them as symmetric adjectives. The analysis uses two distinct datasets: one comprising music generated by humans (referred to as the "human-made" dataset) and the other containing music generated by a machine learning model conditioned by textual descriptions (referred to as the "AI-generated" dataset). The human dataset includes tracks annotated by human evaluators, while the AI-generated dataset's labels are based on semantics-driven input prompts used for generating the music.

We structured the input prompts based on Google's AudioSet ontology [13], which offers a comprehensive framework for categorizing audio. We adopted Russell's circumplex model and chose four emotional adjectives representing the four quadrants of the valence/arousal plane.

Our analysis aims to delineate relationships between these emotional states across differential adjectives conceptualized as three types of 2D symmetries: diagonal, horizontal, and vertical. For each symmetry, two pairs of emotions were analyzed: for the diagonal symmetries (different valence and arousal), the pairs Happy-Sad and Tender-Angry were analyzed; for the horizontal symmetry (different valence, same arousal), the pairs Happy-Angry and Tender-Sad, and for the vertical symmetry (same valence, different arousal) the pairs Happy-Tender and Sad-Angry (see Figure 2).

2.1 Databases

The AI-generated dataset was obtained using audios produced by MusicGen, an artificial intelligence model developed by Meta for generating music [7]. The model was used to create musical compositions based on text descriptions following Google's AudioSet ontology.

Prompts for the audio generation were created using the following phrase structure: - "A (mood) (genre) (role) (concept)" - with four variables "mood", "genre", "role", and "concept" (e.g., "A happy Rock soundtrack song").

In our study, "concept" and "role" remained constant as "soundtrack songs", while the chosen genres were Rock and Pop. Ten audios, 12 seconds long, were generated for each of the seven proposed emotions in the ontology (Happy, Funny, Sad, Tender, Exciting, Angry, Scary) for each selected genre. The AI-generated dataset comprises 80 musical fragments, 20 per emotion/quadrant, of which 10 were described within the genre of Rock music and the other ten were described within Pop music.

The human database, derived from the 4Q audio emotion dataset [2], includes tracks from AllMusic re-categorized in terms of Russell's quadrants [14].

Since not all adjectives proposed by the Google ontology were presented as tags in AllMusic musical fragments, we reduced our analysis to labels: Happy, Sad, Gentle, and Angry, ensuring one emotion per quadrant and assuming the adjectives gentle and tender as equivalent for the purposes of our analysis. From these, 20 samples per emotion, were randomly selected, leading to a human dataset comprised of 80 Rock and Pop music fragments, 20 per emotion/quadrant.

2.2 Features Selection

Based on Livingstone's music-emotion rules [8] and Panda's eight musical dimensions description [12], we selected seventeen musical features organized into a reduced

[2] Webpage: https://mir.dei.uc.pt/downloads.html

Table 1. Selected Features for each Musical Dimensions

Musical Dimension	Selected Features
Melody	Pitch Salience Mean
	Pitch Salience Stdev
Harmony	Chords Number Rate
	Chords Change Rate
	Dissonance Mean
	Dissonance Stdv
	HPCP Entropy Mean
	HPCP Entropy Stdev
	Scale (Krumhansl)
Rhythm	Tempo (BPM)
	Tempo Histogram Entropy
	Rhythm Onset Rate
Dynamic	Average Loudness
Tone Color/Timbre	Spectral Kurtosis Mean
	Spectral Kurtosis Stdv
	Spectral Centroid Mean
	Spectral Centroid Stdv

set of five categories (See Table1):

Melodic Features: A successful strategy for melody extraction derived from the salience-based methods [15]. We include Pitch Salience (mean and standard deviation) to capture the perceptual prominence of pitch.

Harmonic Features: Harmonic Pitch Class Profile Entropy (mean and standard deviation) quantifies the diversity of harmonic content, Chord Changes Rate assesses the dynamism in harmony, and Scale (major or minor) provides insights into the tonal context.

Additionally, the chord number Rate evaluates the harmonic variety, all of which might give insights into emotional nuances in music.

Rhythmic Features: Onset Rate Measures and Tempo (including Tempo Histogram Entropy) provide insights into the structure and pace of the music, which might reflect its energetic and, consequently, emotional states.

Dynamic Features: Average Loudness is considered to gauge the intensity level of the music, offering clues about its emotional impact.

Tone Color/Timbre Features: Spectral Centroid and Spectral Kurtosis (both mean and standard deviation) alongside Roughness/Sensory Dissonance explores the sound's texture and quality, which are crucial for emotional coloring.

Several algorithms have been proposed for audio analysis in music information retrieval. We used Essentia library [16] to compute the selected features.

2.3 Statistical Analysis

To assess the significance of the selected musical features across the possible emotional pairs, we employed various statistical tests tailored to the nature of the features under examination.

For continuous features, the Shapiro-Wilk test was initially conducted to assess normality, accompanied by Levene's test for evaluating equality of variances across groups. If both normality and equality of variances criteria were checked, the Student's t-test was utilized to determine the statistical significance of differences between groups. In cases where data passed the normality test but failed the equality of variances test, Welch's t-test was applied to account for variance discrepancies. For data not satisfying either normality or equality of variances, the non-parametric Mann–Whitney U test was selected to compare differences between groups.

The Chi-square test was deployed for categorical features to assess statistically significant differences among each pair of emotional states.

The significance value, or p-value, derived from the Student's t-test, Welch's t-test, Mann–Whitney U test, and Chi-square test, quantifies the probability that the observed differences between groups occurred under the null hypothesis of no actual difference. A lower p-value suggests a higher statistical significance, indicating strong evidence against the null hypothesis and implying that the observed differences are likely attributable to the variables under investigation rather than chance.

Across all tests, a significance level (α) of 0.05 was adopted, indicating that results with a p-value less than 0.05 were considered statistically significant, reflecting a less than 5% probability that observed differences occurred by chance.

3. RESULTS AND ANALYSIS

We present the findings of our study, analyzing the results derived from both human and AI-generated datasets [3].

The significance values across the datasets are presented in Figure 3. These values are depicted as the confidence value, $1 - p$-value, to represent the significance more intuitively; higher values thus indicate greater statistical significance.

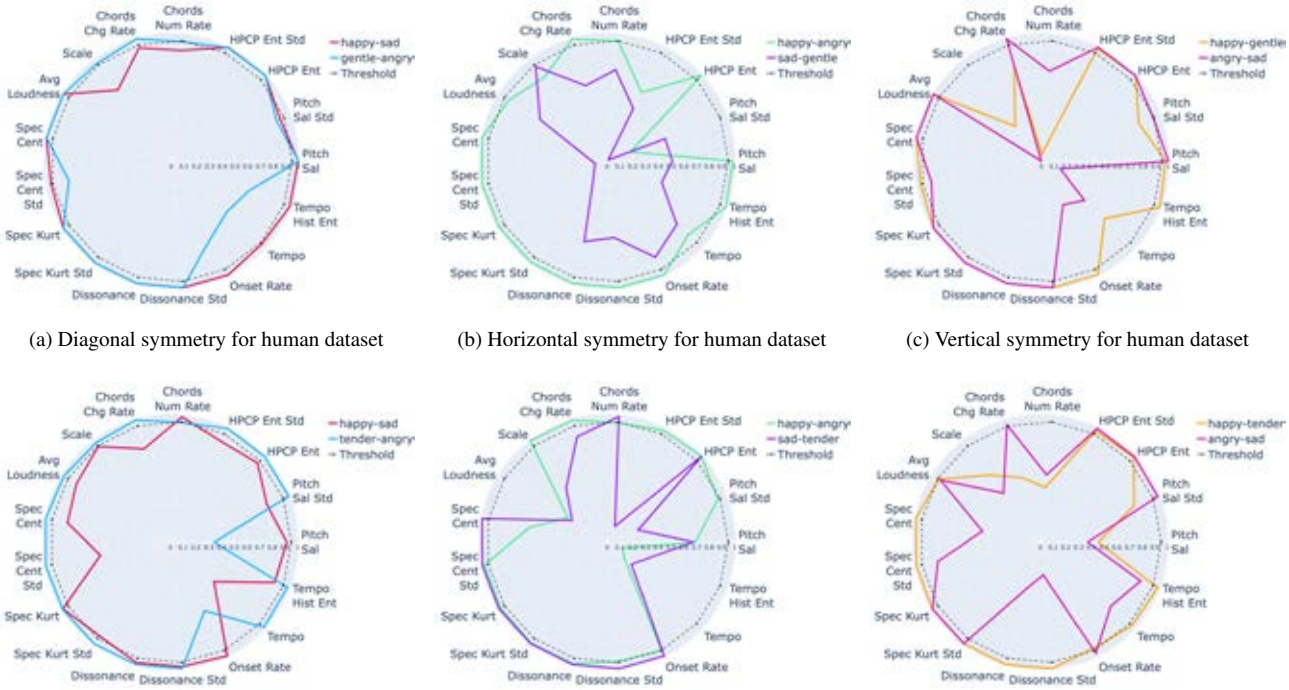
3.1 Diagonal symmetries

In analyzing the diagonal symmetries, we searched for contrasts in both valence and arousal dimensions. This approach entails comparing "happy" with "sad" and "tender/gentle" with "angry". Significance values derived from this analysis are illustrated in Figure 3a for the human dataset and Figure 3d for the AI-generated dataset.

3.1.1 Happy-Sad

The human dataset shows significant differences across features within the melodic, harmonic, rhythmic, and tim-

[3] Code for the data processing and feature extraction, statistical analysis, and further results can be found at `https://github.com/braga1376/musical-features-emotional`.



(a) Diagonal symmetry for human dataset (b) Horizontal symmetry for human dataset (c) Vertical symmetry for human dataset

(d) Diagonal symmetry for AI-generated dataset (e) Horizontal symmetry for AI-generated dataset (f) Vertical symmetry for AI-generated dataset

Figure 3. Comparative analysis of the confidence value ($1 - p$-value) across different movements for human and AI-generated datasets. From left to right: diagonal, horizontal, and vertical movements.

bral musical dimensions that effectively distinguish the emotion expressed by music. These differences highlight "happy" music's diverse chord usage, higher loudness levels, and faster tempo, as well as brighter, more complex, and rougher timbre compared to "sad" music.

For the AI-generated dataset, the analysis shows differences across features in the harmonic, rhythmic, and tone color/timbral categories. "Happy" music exhibits a greater diversity of chords and more frequent harmonic changes than "sad" music, indicating a richer harmonic complexity.

3.1.2 *Tender/Gentle-Angry*

Considering the human dataset, our results show statistical differences across all musical dimensions under analysis except rhythmical. The results indicate "angry" music's tendency towards greater harmonic complexity, the use of the minor scale, as well as brighter, more complex, and rough sound texture when compared to "gentle" music.

The AI-generated dataset has pronounced differences across all musical dimensions. These differences denote "angry" music's diverse chord usage, fast harmonic changes, and faster tempos, contrasting with "tender" music's less varied harmony and more constant timbre.

3.2 Horizontal Symmetry

Horizontal symmetry contains emotional adjectives with the same arousal but opposite valence within the Russell space. This symmetry compares high-arousal emotional descriptors as "happy" and "angry" and low-arousal descriptors as "sad" and "tender/gentle'. The findings from this comparative study are depicted in Figure 3b for the human dataset and Figure 3e for the AI-generated dataset.

3.2.1 *Happy-Angry*

Significant differences were found in the human dataset for the "happy-angry" pair across all musical dimensions except for dynamics. These differences suggest "angry" music's broader range of chords and more frequent chord transitions, brighter timbre, and greater dissonance, as well as less frequent tempo changes and greater tempo variability compared to "happy" music.

In the AI-generated dataset, the differentiation between "happy" and "angry" music primarily focuses on harmonic, rhythmic, and timbral features. At the same time, the melodic and dynamic dimensions show less significant features that differentiate both groups. The results indicate that "angry" music has a more dynamic harmony and greater chord diversity, often utilizing the minor scale, in contrast to the predominantly major scale of "happy" music, with more frequent changes, brighter spectral range, and increased auditory dissonance.

3.2.2 *Sad-Tender/Gentle*

Considering the human dataset for the "sad" and "tender" pair, no feature shows statistical significance, highlighting a possible similarity between the music that expresses the mentioned emotions.

The AI-generated dataset, in turn, denotes significant differences predominantly in features belonging to harmony, rhythm, and timbre dimensions. These differences indicate that "sad" music has a more diverse chord usage, less

frequent changes, and brighter timbre when compared to "tender" music.

3.3 Vertical Symmetry

In the vertical symmetry analysis, we explored emotional adjectives with similar valence but opposite arousal levels within the Russell space. This involves the comparative analysis between the positive valence adjectives "happy" and "tender/gentle" and the negative valence emotions "angry" and "sad". Results from this comparative analysis are presented in Figure 3c for the human dataset and Figure 3f for the AI-generated dataset.

3.3.1 Happy-Tender/Gentle

In the analysis of the human dataset for the "happy-gentle" pair, all categories of musical features show significant differences. The separability of the musical features highlights that "happy" music has a more diverse chord usage, brighter timbre, higher auditory dissonance, more frequent changes, and lower tempo variability.

In the AI-generated dataset, the distinction between "happy" and "tender" music is marked by significant differences in features in the timbral, harmonic, and rhythmic musical dimensions. These differences suggest "happy" music's more diverse chord usage, higher tempo, and brighter timbre.

3.3.2 Angry-Sad

Considering the human dataset, significant differences across melody, harmony, dynamics, and timbre features permit the distinction between "angry" and "sad" music. These results point to "angry" music's more diverse chord usage, brighter timbre, and increased auditory dissonance compared to "sad" music.

The AI-generated dataset also has significant differences across the melodic, harmonic, rhythmic, and timbral dimensions but has fewer features in each category. These suggest "angry" music's more diverse and frequently changing chords as well as more frequent changes.

4. DISCUSSION

We evaluated our findings according to Livingstone's framework on musical emotion featuring. Afterward, we compare human and AI-generated datasets regarding semantic differentiation through a set of musical features.

Our findings broadly agree with Livingstone's music-emotion rules, predicting differences across all musical characteristics for the diagonal symmetries analysis.

Specifically, for the "happy-sad" comparison, significant differences are observed in all musical feature categories, with more pronounced differences in the human-generated dataset. Similarly, the "tender/gentle-angry" comparison reveals significant differences across most musical dimensions. However, an exception arises with rhythm in the human dataset, where no significant differences are found, diverging from Livingstone's framework.

Our results intersect with and diverge from Livingstone's studies regarding horizontal symmetry, similar arousal levels, and opposite valence values.

For the "happy-angry" pair, our analysis aligns with Livingstone's hypothesis regarding dynamics and tempo consistency, as no significant differences in tempo were found. Some harmonic features significantly differed, supporting expected variances in harmony complexity. However, significant differences in timbre diverge from Livingstone's model.

The "sad-tender/gentle" pair further complicates the picture. The human dataset's tendency towards significant difference in scale aligns with expected mode consistency, yet harmony complexity remains unchanged. The AI-generated dataset, however, shows significant timbre differences and rhythmic distinctions through Onset Rate, not Tempo, diverging from Livingstone's model.

Results on the vertical symmetry partially align with Livingstone's findings.

For the "happy-tender/gentle" comparison of high valence emotions, significant differences are noted across all musical dimensions in the human dataset and into the timbral and rhythmic categories in the AI-generated dataset, differing from Livingstone's expectations by revealing distinctions in harmony complexity.

However, the expected consistency in scale is observed, with no significant differences in this feature. Similarly, the "angry-sad" comparison, reflecting low valence emotions, shows a broad feature differentiation in the human dataset and more constrained distinctions in the AI-generated dataset, corresponding to the expected scale consistency. Despite this, differentiation in harmony complexity and minimal rhythmic distinction contradicts Livingstone's model.

When comparing human and AI-generated datasets, the diagonal symmetries reveal a minimal divergence between these, although more differentiation is observed in the human dataset. This alignment suggests that although both sources adequately separate these emotions, the human database has a more robust distinction for the same musical features, especially in the happy-sad emotional pair.

The horizontal symmetry shows distinctions in the human dataset influenced by arousal intensity. High arousal conditions yield significant differentiation, whereas low arousal conditions do not. The AI-generated samples display a uniform response considering the arousal value. The number of significant features in the timbre categories increases notably for the "sad-tender" emotional pair compared to the human database.

In vertical symmetry comparison, the human dataset does not depend on the valence for distinguishing emotions, unlike the AI-generated dataset, which exhibits dependence on valence. Specifically, high valence in the AI-generated dataset shows greater differentiation than low valence, indicating the capability to generate music more distinguishable regarding valence.

5. CONCLUSIONS

Although the semantic representation of emotions is highly subjective, this study gives insights into the relationship between musical features and emotional states across human and AI-generated Rock/Pop music, focusing on diagonal, vertical, and horizontal differences within the valence/arousal space.

Our findings indicate significant separability for diagonal symmetry pairs in the human dataset, whereas the AI-generated music exhibits clear separability primarily in the angry vs. tender pair. Additionally, we inferred that the level of arousal influences the separability of horizontal symmetries for the human dataset and valence influences, although less, the separability of vertical symmetries for the AI-generated dataset.

We also conclude that our work aligns with Livingstone's regarding diagonal comparison and, to some extent, regarding horizontal and vertical. Still, it is essential to note the ability of the text-to-music model MusicGen [7] to generate notably different music when comparing the semantics from the prompt and mostly aligned regarding the valence/arousal comparisons.

Thus, this investigation contributes to the field by highlighting the differences in emotional perception between human and AI-generated music, offering a novel methodology for future research in music information retrieval and the analysis of emotion.

Further research could expand on two strands. First, the musical features used could be extended and further studied, in line with work on musical features for emotion recognition [12], to better understand the part played by the different musical categories and make a more robust analysis. Second, it could expand on the comparison between human and AI-generated music across various genres and emotional states, contributing to the evolving dialogue between musicology, psychology, and artificial intelligence.

Acknowledgments

The authors would like to thank Nuno Correia and Roger B. Dannenberg for their input on this work. This work was partially funded by Portuguese national funds through Fundação para a Ciência e Tecnologia (FCT) under CMU Portugal, by the Ph.D. scholarship with references PRT/BD/154690/2023 and 2022.11918.BD

6. REFERENCES

[1] P. Ekman, "Universal and cultural differences in facial expression of emotions," *Nebraska Symposium on Motivation*, pp. 207–283, 1972.

[2] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[3] I. B. Mauss and M. D. Robinson, "Measures of emotion: A review," *Cognition and emotion*, vol. 23, pp. 209–237, 03 2009.

[4] C. E. Osgood, G. J. Suci, and Percy H. Tannenbaum, *The Measurement of Meaning*. University of Illinois Press, 1957.

[5] J. Forero, G. Bernardes, and M. Mendes, "Are words enough?" *AIMC 2023*, aug 29 2023, https://aimc2023.pubpub.org/pub/9z68g7d2.

[6] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi, M. Sharifi, N. Zeghidour, and C. Frank, "Musiclm: Generating music from text," 2023.

[7] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[8] S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson, "Changing musical emotion: A computational rule system for modifying score and performance," *Computer Music Journal*, vol. 34, no. 1, pp. 41–64, 2010.

[9] E. Schubert, "Measurement and time series analysis of emotion in music," 1999.

[10] A. Gabrielsson and E. Lindstrom, "The influence of musical structure on emotional expression." 2001.

[11] A. Gabrielsson and P. N. Juslin, "Emotional expression in music." pp. 503–534, 2001.

[12] R. Panda, R. Malheiro, and R. P. Paiva, "Audio features for music emotion recognition: a survey," *IEEE Transactions on Affective Computing*, vol. 14, no. 1, pp. 68–88, 2020.

[13] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[14] R. Panda, R. Malheiro, and R. P. Paiva, "Novel audio features for music emotion recognition," *IEEE Transactions on Affective Computing*, vol. 11, pp. 614 – 626, 03 2018.

[15] J. Salamon and E. Gomez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[16] D. Bogdanov, N. Wack, E. Gómez Gutiérrez, S. Gulati, H. Boyer, O. Mayor, G. Roma Trepat, J. Salamon, J. R. Zapata González, X. Serra *et al.*, "Essentia: An audio analysis library for music information retrieval," in *Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); p. 493-8.* International Society for Music Information Retrieval (ISMIR), 2013.

TEMPORAL ANALYSIS OF EMOTION PERCEPTION IN FILM MUSIC: INSIGHTS FROM THE FME-24 DATASET

Ruby O. CROCKER[1] and **György FAZEKAS**[1]

[1] *Center for Digital Music*, **Queen Mary University of London**, London, England

ABSTRACT

Understanding how composers evoke emotions through music remains limited due to the lack of diversity in existing datasets used for computational analysis of film music. To address this gap, the Film Music Emotion Dataset (FME-24) was created to explore emotion perception in modern and contemporary film music, covering various genres and compositional styles. FME-24 introduces precise temporal control through comprehensive time stamps, enabling nuanced analyses often overlooked in previous studies. Over 185 participants contributed annotations for the dataset. While there is consensus on arousal, there's significant valence variation across samples, possibly due to temporal aspects and participant differences. On average, each participant completed 17 annotations across 10 randomly generated samples, totaling 1580 annotations. These annotations will help uncover parallels between emotional states and musical transitions, enriching our understanding of music's impact on audience immersion. Initial findings indicate potential for meaningful insights, future work including extracting musical features to compare with valence-arousal values and expanding the sample size with additional participants will enhance the reliability of the results, and uncover patterns between emotion perception and musical elements.

1. INTRODUCTION

Composers and filmmakers use music to manipulate emotion to aid storytelling. Changes in music based on characteristics or elements by composers can alter the way in which visual scenes are interpreted, film music can differ from mainstream or commercial music as there can be alternative motivations for the direction of the composition. Juslin and Sloboda [1], propose that film music stands out as one of the most powerful tools for conveying emotion in cinematic experiences.

Juslin and Laukka [2] proposed that one of the most prevalent reasons for listening to music is emotion. It is suggested that understanding the human listening response to music in everyday context provides a better understanding of how emotion in music is perceived. There is evidence indicating distinctions between emotions evoked by music and those experienced in everyday life [3].

The unique ability of film music to replicate or evoke emotions through immersion makes it an intriguing subject for exploration, especially considering the distinctions observed between music-induced emotions and those experienced in everyday life [4]. There is recent research based around composition and film music being the carrier of emotion in film [5], however there is minimal research focusing on how composers manipulate emotion based on musical attributes and techniques.

The distinction between emotion expressed by a piece of music and the emotion felt by a listener is an essential part of research for emotion in music research [6]. Both perceived and felt emotional responses are relevant for studies on emotional responses to music [7]. There is a disparity between the emotional response perceived by listeners and the emotions they feel. These discrepancies may be attributed to various factors, including contextual information, memory, physiological arousal, age, and musical features such as tempo and suggested emotions [6].

2. RELATED WORK

Wei et al. [8] explored the impact of music on perceived emotions in film and stated that the impact of film music on viewers' emotions is still not fully understood. An experiment was conducted to analyse how music influences emotion perception in film, they compared viewers responses with video only, audio only, and both video and audio, they had 16 clips that covered 4 film genres. The focus was on the influence of both image and sound. There is limited research in how mood is measured over-time in film music and how these emotions change and develop over-time.

Russell [9] proposed a valence-arousal model of emotion, Figure 1, offering a dimensional framework for understanding emotional experiences. This model, widely adopted in emotion research, provides detailed insights into emotional nuances compared to categorical approaches [10]. Arousal in this model signifies the energy level of an emotion, while valence indicates its positivity or negativity.

Existing film music datasets predominantly focus on classical and mainstream music, lacking representation of modern genres and arthouse cinema [11, 12]. There is a scarcity of film music datasets from the past two decades, particularly those reflecting diverse cultural backgrounds and cinematic styles [13].

Walus [14] explored approaches to composing film music, emphasizing the dynamic nature of scoring during pre-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Figure 1: Russell's Arousal-Valence Emotion Model

production. Changes in the score's layers can significantly impact narrative, character development, and mood in the film.

To draw meaningful conclusions about perceptions of film music, it is crucial to analyze a diverse array of compositions spanning different genres, styles, cultural backgrounds, and cinematic contexts [15].

The paper proceeds as follows: Section 3 introduces the Film Music Emotion Dataset (FME-24), outlining its creation process and characteristics. Section 4 analyzes the demographics, inter-rater agreement, and initial evaluation of FME-24. Initial findings are discussed in Section 5, followed by future research directions in Section 6.

3. FME-24 DATASET

FME-24 dataset was developed to address the gaps in existing research explained in Section 1. FME-24 was created to study composers' techniques for manipulating emotions in film music across different styles. The dataset is populated with annotations from various participants to provide valence-arousal annotations with corresponding timestamps within film compositions. The initial purpose of creating the dataset was to identify music features that coincide with participants' emotional perceptions, aiming to pinpoint specific characteristics during moments of emotional change.

The dataset consists of 300 compositions from 21 different film genres with over 156 different composers. Many films in the dataset are classified under a blend of multiple genres. The predominant film genre label in the dataset, occurring 229 times, is 'Drama.' This predominance stems from the frequent combination of 'Drama' with other genre labels within many film genres. The other most common film genres in the dataset consist of; Action: 51, Adventure: 64, Biography: 38, Comedy: 67, Romance: 42, Crime: 38, Mystery: 39, Sci-Fi: 26 and Thriller: 27. A combination of multiple genres are used to describe each film. All film metadata, including film genre classifications, release dates, directors, composers, and other relevant details, were sourced from the IMDB database website [1] .

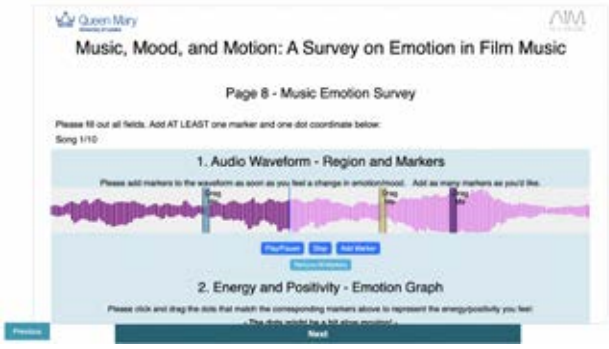
The audio files selected were from a combination of art-house cinema and mainstream cinema from the last twenty years. While the dataset primarily concentrates on music produced within the last two decades, a slight emphasis is placed on compositions from more recent scores due to the increased variety observed in genres and styles during this period. The compositions were selected from award-winning films, regardless of whether the film's soundtrack itself won an award. This decision was made because the focus is on examining the immersive mood created within the film. Award-winning films often undergo rigorous evaluation processes by industry professionals, ensuring a high standard of emotional impact and artistic merit. Selection criteria also included critical acclaim and the capacity to evoke a wide spectrum of emotions, ensuring a diverse range of compositions for comprehensive emotional analysis. In many cases, the emotional impact of a film extends beyond its original soundtrack. It's worth noting that not all films have original scores; some utilize music from older films or mainstream music to enhance mood and atmosphere. In the selection process for creating the dataset, music from sources other than original scores was included, enabling a broader exploration of musical influences on perceived emotions in film music. While the film's soundtrack may not always be recognized with awards, it remains pivotal in evoking emotion within the film. Consequently, if a film has received awards or nominations, it is deemed valuable for inclusion in the dataset. During the selection of compositions for the dataset, discussions were conducted with both film and music students to identify impressive and dynamic scores. When compiling the audio files for the dataset, consideration was given to acclaimed and highly rated films from various prestigious film festivals. The aim was to incorporate a diverse selection of films beyond Western cinema, ensuring a rich representation of cultural influences and styles.

3.1 Dataset Creation and Process

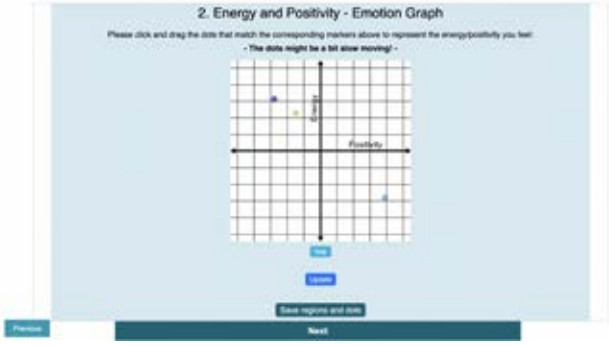
Annotations from participants were collected using an online interactive survey. The interactive space consisted of a playable waveform and an interactive valence-arousal space, shown in Figure 2. Participants were presented with a 15-second excerpt from a film soundtrack, devoid of any visual accompaniment, and solely consisting of the audio waveform. It was explicitly communicated to the participants at the start of the study that the investigation solely related to the perception of emotion in music. Unlike, Wei et al. [8], this paper focuses solely on sound, not including image or visual stimuli. The decision to focus on emotion perception rather than induced emotions is based on the understanding that participants may find it easier to discern and articulate their own perceived emotions. In large-scale data collection, involving participants in self-analysis offers valuable insights and is more efficient, thus encouraging greater participation. By focusing on perceived emotions, it becomes easier to systematically observe and interpret emotional responses, ideal for scientific and analyt-

[1] https://www.imdb.com/

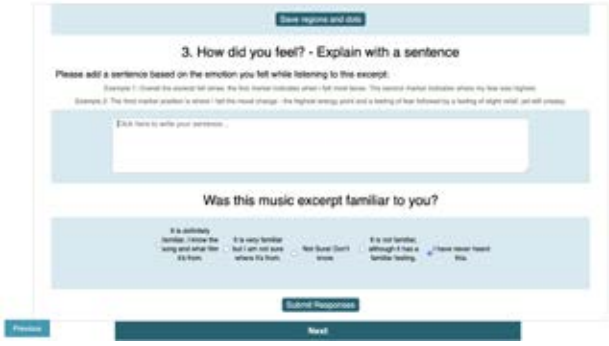
ical contexts, ensuring comprehensive and impartial analysis. Participants were instructed to click a button upon perceiving an emotion change while listening to the excerpt. The instruction was to: "add markers to the waveform as soon as you feel a change in emotion/mood. Add as many markers as you'd like". Simultaneously, a dot appeared on the valence-arousal space in a color matching the marker, which participants could move. Valence and Arousal were labelled 'Positivity' and 'Energy' respectively to ensure understanding for participants. Participants were able to manipulate markers on the waveform and adjust the position of dots on the valence-arousal space. They were then prompted to add a sentence describing their feelings and familiarity rating before submitting their response. Participants were instructed to complete the entire survey, which comprised of 10 excerpts, each lasting 15 seconds. However, they had the option to stop at any time. The 10 excerpts were randomly loaded from the collection of 300 audio files for each participant as they initiated the survey. Additionally, the 15-second samples were randomised for each participant. Participants were encouraged to look at short video before starting the survey to help direct how to navigate the interface, additionally there was a pop-up diagram for further help with interpreting the valence-arousal space. Participants were recruited by distributing the survey as a website link through GitHub Pages [2] and sharing it via targeted mailing lists within musicology, film, and emotion-related fields. Academic mailing lists were also utilized to ensure a broad participation base. While the study focused on film and music, it did not necessitate participants to be musicians or experts in the field. Hence, social media platforms were employed to reach a wider audience. This approach aimed to gather a diverse range of responses from participants with varied backgrounds, including those in psychology, film studies, music, production and related academic disciplines. Ethical approval for this study was obtained, ensuring participant safety and data confidentiality [3] .



(a) Online study - Visual waveform with interactive markers



(b) Online study - Valence-Arousal interactive graph



(c) Online study - Emotion sentence and familiarity rating

Figure 2: Screenshot of online annotation collection study with interactive interface

4. ANNOTATION PROCESS

4.1 Demographics

The predominant age bracket among participants fell within the range of 25 to 34 years old. There were 17 different nationalities and 22 countries given in the demographics form. The majority of participants were from the UK and Europe. There were 62 participants who identified as female, 67 participants who identified as male, 13 provided a different identification or left the field blank and 43 had errors. In regard to employment status, the majority of participants were either employed full-time or enrolled in university. The distribution of age groups and employment status among the 130 participants who took part in the study, after removing errors, is illustrated in Figure 3 and Table 1, respectively. It is clear from the results there is a gap in age range representation, more participants are needed.

4.2 Annotations

FME-24 consists of 300 compositions, with each composition featuring two randomly selected excerpts, thus totaling 600 samples in the dataset. These samples were annotated by 185 different annotators, each annotator provided annotations for 10 audio samples. Overall, there are 1580 individual annotations across the entire dataset. After eliminating errors, the average number of annotations per sample ranges from 4 to 5.

The number of annotations per sample varied among participants, ranging from a minimum of 1 annotation to a maximum of 18. The annotations consist of time stamps where participants felt a change in mood, correspond-

[2] https://rubycrocker.github.io/film-score-emotion-survey/

[3] The study was conducted in accordance with the guidelines and regulations provided by The School of Electronic Engineering and Computer Science Ethics Approval Committee at Queen Mary University, with reference number: QMERC20.565.DSEECS23.131 - Music, Mood, and Motion: A Survey on Emotion in Film Music'.

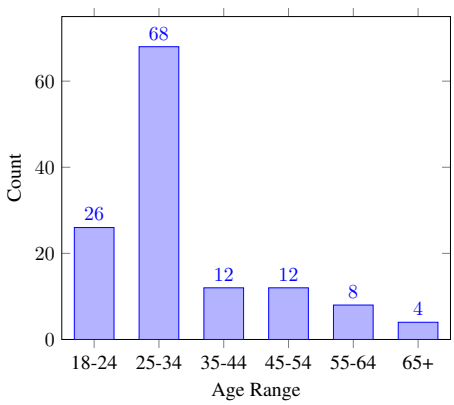


Figure 3: Distribution of Age Groups

Employment Status	Count
Full-time employment	**42**
Full-time parent	2
Other	6
Part-time employment	24
Retired	4
Self-employment	22
University	**44**

Table 1: Distribution of Employment Status

ing coordinates that represent the energy and positively felt at that time point in valence-arousal space, an emotion sentence from the participant about the excerpt, and a familiarity rating each. After filtering out entries containing errors and refining the dataset by eliminating entries with only one annotator, as well as excluding participants self-describing limited music interest or experience from the Goldsmiths Musical Sophistication Index (Gold-MSI) [4] assessments, the validation process confirmed 192 samples. While the study initially involved 185 annotators, subsequent error adjustments resulted in a final count of 130 participants. Increasing the number of participants and annotations would enhance the reliability of the dataset, as the numbers are significant but considerably insufficient for the scope of the data. Table 2 shows the summary of statistics in the dataset before and after refining the dataset and removing errors. Although this refinement reduces the size of annotations, participants and samples it enhances the overall quality and reliability of the data, ensuring more accurate and meaningful analysis.

	Original	Refined
Participants	185	130
Soundtracks	278	161
Samples	439	192
Total Annotations	1580	922
Annotations per Sample	3-4	4-5

Table 2: Summary statistics of the FME-24 dataset

[4] https://www.gold.ac.uk/music-mind-brain/gold-msi/

4.3 Inter-rater Agreement

	Arousal	Valence
Range	0.418103	0.435423
IQR	0.208950	0.218257

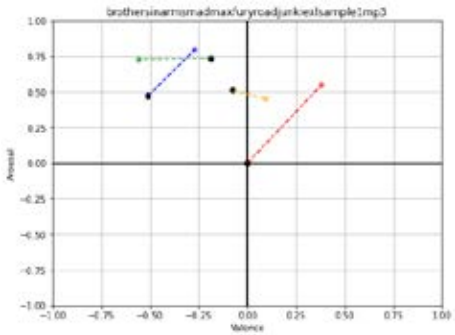
Table 3: Range and Inter-Quartile Range (IQR) for Arousal and Valence

Table 3 shows the range and Inter-Quartile Range (IQR) for Arousal and Valence. The larger range in valence compared to arousal indicates greater variability in valence perceptions. This suggests that individuals typically encounter greater difficulty in perceiving valence compared to arousal. This observation is consistent with findings from previous research studies [16–18]. The dataset exhibits considerable variation in inter-rater agreement, suggesting that some songs are more consistently perceived in agreement than others. This could be due to many factors including; cultural differences, subjective nature of emotions, music and film, familiarity of the stimuli, personal preferences, music ability, contextual factors - the ambiguity of the story with it being sound only. During the online survey Gold-MSI was used to analyse music ability and preferences of the participants. This approach facilitates the identification of participants with higher levels of musical ability and facilitates further analysis of their responses.

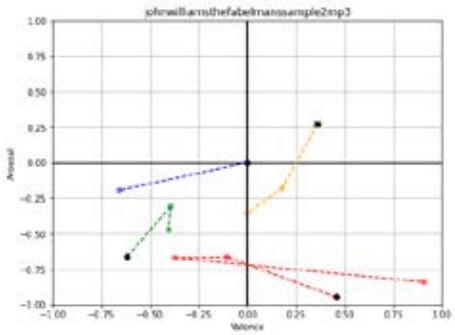
5. DISCUSSION

Plotting the distribution of participant annotations reveals a notable consensus in arousal levels, see Figure 4, indicating a general alignment in the perceived energy across responses. While there might be minor discrepancies in valence interpretations, the overall trend suggests a shared understanding of the emotional intensity conveyed by the music. This observation is interesting in terms of film music as it highlights composers' ability to effectively evoke a consistent energetic atmosphere, even if participants have subjective responses to the positivity and negativity in music.

Figure 4 illustrates the emotion perception over time in the valence-arousal space of two audio files assessed by four different annotators. Upon visual inspection, it becomes apparent that there was a degree of consensus among these raters. While the plotted points are not densely clustered, they are not excessively sparse either, suggesting a moderate level of agreement. However, it is worth noting that the initial impression based solely on the first plotted point may lead to a different interpretation. The temporal aspect of the plotting may give the appearance of sparsity, but this could be attributed to annotators emphasizing perceptual changes rather than a lack of diversity. The black marker represents the first time point added by a participant, while the dotted lines show the progression to subsequent change points. Despite these variations, a consistent trend emerges, suggesting relative agreement on arousal or valence. Most plots exhibit clustering in either the top or bottom half of the valence-arousal space,

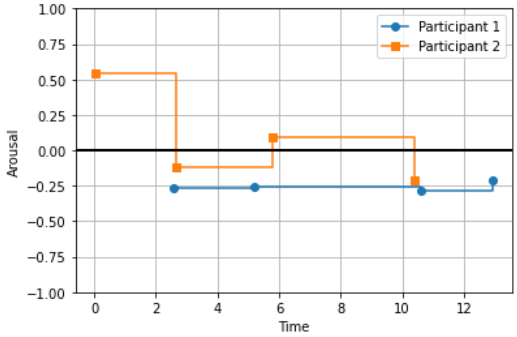


(a) AV participant annotations over-time for *'Brothers In Arms'*

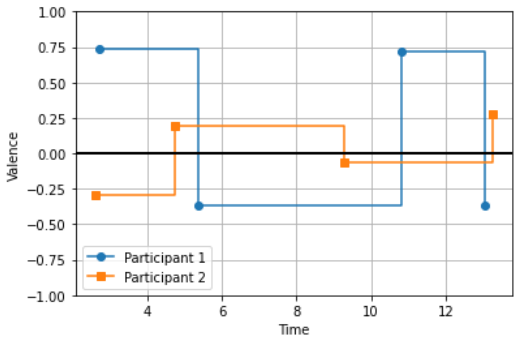


(b) AV participant annotations over-time for *'The Fabelmans'*

Figure 4: AV participant annotations over-time



(a) Arousal Over time for 2 participants for *'Remains - Volker Bertelmann'*



(b) Valence Over time for 2 participants for *'Remains - Volker Bertelmann'*

Figure 5: AV participant annotations over-time for *'Remains - Volker Bertelmann'*

suggesting a consensus on arousal perception, even if disagreement in valence perception, this is also shown in Figure 4.

Figure 5 shows how 2 participants perceive arousal and valence over-time in the composition: *'Remains - Volker Bertelmann'* from *All Quiet on the Western Front*. The two participants seem to mostly agree on the time points where they perceive a change, in arousal there appears to be a lot of agreement but for valence the perceptions are almost opposite. This is interesting because the time-points are still relatively close together but complete disagreement in positivity and negativity but agreement in energy levels in the excerpt.

Looking at the distribution of annotations across genres within the dataset it is clear that participants seem to plot significantly more emotion ratings in the top half of the arousal-valence graph. This is shown in Table 4. These values are calculated from the refined dataset. Each column represents a quadrant of the arousal-valence plain, many of these genres are combined with other genres. The positive arousal areas have the highest number of annotations.

6. CONCLUSIONS AND FUTURE WORK

In summary, the FME-24 dataset is the first film composition dataset with valence-arousal annotations and timestamps. It will help provide insights on how individuals perceive arousal and valence in music. This comprehensive dataset, curated from diverse cultural sources and film festivals, reveals a remarkable consensus in the perceived energy levels of compositions. These findings underscore the dataset's significance in advancing our understanding of music's emotional impact and its potential for future research. The FME-24 dataset serves as a valuable resource for exploring the intricate relationship between music and emotion for various contexts.

While the dataset comprises a substantial number of data points and participants, it's important to acknowledge that the available information per track is relatively limited at this stage. The current participant count is significant but insufficient for the study's scope. Increasing the sample size would enhance reliability. Further exploration and analysis may lead to deeper insights into the richness of the data and its potential for drawing robust conclusions. Examining further into the dataset, the aim is to uncover patterns in the distribution of arousal and valence over time. The objective is to identify correlations between shifts in emotional states, regardless of matching arousal and valence values. This involves closely analyzing the temporal aspect of data collection, scrutinizing the proximity of timestamps across different participants using various window intervals.

In future research, musical features will be extracted and compared with valence-arousal values using machine learning techniques. Further statistical analysis will be undertaken to understand how participants perceive changes in emotion over time. Various approaches, such as comparing the distance between arousal-valence points and analyzing emotion sentence data, will be employed to uncover

Genre	+A/+V	-A/-V	+A/-V	-A/+V
Biography	**39**	14	30	11
Comedy	**112**	12	44	27
Crime	28	16	**37**	10
Drama	**291**	122	167	123
History	**14**	0	10	0
Musical	**9**	0	7	0
Animation	**44**	6	14	7
Adventure	**96**	11	56	21
Sci-Fi	28	12	**30**	12
Thriller	26	10	**29**	11
Action	53	17	**64**	12
War	**16**	6	10	5
Romance	**63**	19	16	31
Mystery	48	18	**56**	20
Horror	23	12	**35**	9
Documentary	**9**	2	5	0
Music	**22**	4	9	2
Fantasy	**38**	7	15	11
Family	**10**	1	3	0
Western	**6**	0	0	0
Sport	**6**	4	3	0

Table 4: Distribution of Annotations by Genre Across Valence-Arousal Quadrants

additional insights from the collected data.

In conclusion, the FME-24 dataset is well-positioned to help improve our understanding of the intricate relationship between music and emotion in film, providing a valuable resource for future research and enhancing our comprehension of the profound influence of music on our emotional responses.

Acknowledgments

This work was funded by UK Research and Innovation (UKRI). We acknowledge this research has been supported Mathieu Barthet, Charalampos Saitis and Simon Dixon at Queen Mary University of London (QMUL) for their guidance and assistance throughout this research. I would like to express my gratitude to all the individuals who have participated in the survey so far.

7. REFERENCES

[1] P. N. Juslin and J. Sloboda, *Handbook of music and emotion: Theory, research, applications*. Oxford University Press, 2011.

[2] P. N. Juslin and P. Laukka, "Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening," *Journal of new music research*, vol. 33, no. 3, pp. 217–238, 2004.

[3] P. N. Juslin, *Musical emotions explained: Unlocking the secrets of musical affect*. Oxford University Press, USA, 2019.

[4] S. K. Jensen and N. Salmose, "Media and modalities: film," 2022.

[5] A.-K. Herget, "Well-known and unknown music as an emotionalizing carrier of meaning in film," *Media Psychology*, vol. 24, no. 3, pp. 385–412, 2021.

[6] E. Schubert, "Emotion felt by the listener and expressed by the music: literature review and theoretical perspectives," *Frontiers in psychology*, vol. 4, p. 837, 2013.

[7] P. N. Juslin, G. Barradas, and T. Eerola, "From sound to significance: Exploring the mechanisms underlying emotional reactions to music," *The American journal of psychology*, vol. 128, no. 3, pp. 281–304, 2015.

[8] C. Wei, T. Kronland-Martinet, Y. Frachi, and M. Barthet, "Influence of music on perceived emotions in film," in *Audio Engineering Society Convention 153*. Audio Engineering Society, 2022.

[9] J. A. Russell, "A circumplex model of affect." *Journal of personality and social psychology*, vol. 39, no. 6, p. 1161, 1980.

[10] P. Evans and E. Schubert, "Relationships between expressed and felt emotions in music," *Musicae Scientiae*, vol. 12, no. 1, pp. 75–99, 2008.

[11] M. Richards, "Film music themes: Analysis and corpus study." *Music Theory Online*, vol. 22, no. 1, 2016.

[12] H. D. Jones, "Euro-million arthouse films: Diverse and insightful stories, niche audiences," in *Transnational European Cinema: Representation, Audiences, Identity*. Springer, 2024, pp. 107–140.

[13] F. Morgan and J. Mooney, "'the same trade as mozart': Convincing the sceptics of electronic music's value," *The Journal of Popular Television*, vol. 9, no. 1, pp. 55–77, 2021.

[14] B. P. Walus, "A new modular approach to the composition of film music." Ph.D. dissertation, 2013.

[15] B. Ma, T. Greer, D. Knox, and S. Narayanan, "A computational lens into how music characterizes genre in film," *PloS one*, vol. 16, no. 4, p. e0249957, 2021.

[16] J. Grekow, "Audio features dedicated to the detection and tracking of arousal and valence in musical compositions," *Journal of Information and Telecommunication*, vol. 2, no. 3, pp. 322–333, 2018.

[17] B. Schuller, J. Dorfner, and G. Rigoll, "Determination of nonprototypical valence and arousal in popular music: features and performances," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2010, pp. 1–19, 2010.

[18] H. Egermann, N. Fernando, L. Chuen, and S. McAdams, "Music induces universal emotion-related psychophysiological responses: comparing canadian listeners to congolese pygmies," *Frontiers in psychology*, vol. 5, p. 1341, 2015.

BiSAID: Bipolar Semantic Adjectives Icons and Earcons Dataset

Zijing Cao (up202001544@fe.up.pt) (0009-0000-0214-4436)[1],
António S. Pinto (asapinto@fe.up.pt) (0000-0003-1629-8385)[1], and
Gilberto Bernardes (gba@fe.up.pt) (0000-0003-3884-2687)[1]

[1]**University of Porto – Faculty of Engineering & INESC TEC**, Porto, Portugal

ABSTRACT

This paper presents BiSAID, a dataset for exploring bipolar semantic adjectives in non-speech auditory cues, including *earcons* and *auditory icons*, i.e., sounds used to signify specific events or relay information in auditory interfaces from recorded or synthetic sources, respectively. In total, our dataset includes 599 non-speech auditory cues with different semantic labels, covering temperature (cold vs. warm), brightness (bright vs. dark), sharpness (sharp vs. dull), shape (curved vs. flat), and accuracy (correct vs. incorrect). Furthermore, we advance a preliminary analysis of brightness and accuracy earcon pairs from the BiSAID dataset to infer idiosyncratic sonic structures of each semantic earcon label from 66 instantaneous low- and mid-level descriptors, covering temporal, spectral, rhythmic, and tonal descriptors. Ultimately, we aim to unveil the relationship between sonic parameters behind earcon design, thus systematizing their structural foundations and shedding light on the metaphorical semantic nature of their description. This exploration revealed that spectral characteristics (e.g. spectral flux and spectral complexity) serve as the most relevant acoustic correlates in differentiating earcons on the dimensions of brightness and accuracy, respectively. The methodology holds great promise for systematizing earcon design and generating hypotheses for in-depth perceptual studies.

1. INTRODUCTION

In sound-related fields, the formal discourse by professional sound designers and musicians often use sensory metaphors as a practical language to describe the attributes of sound. They draw from a myriad fields, such as visual and haptic, encompassing semantic adjectives such as high, low, deep, warm, cold, bright, dark, rough, smooth, big, small, soft, edgy, round, flat, sharp, dull, transparent, translucent, shimmering, sweet, and colorful. These metaphors play a crucial role in shaping the *meaning* of sonic objects and narratives through their crossmodal relationships. For instance, in film sound design, weapons like swords, knives, or daggers are consistently accompanied by high-pitched, sharp sounds, aligning with their semantic characteristics, even if they do not precisely mimic the physical reality [1].

The discourse on the communicative potential of sound has recently gained traction in the field of auditory displays, namely on what concerns non-verbal auditory cues, which can be categorized into two primary types: auditory icons and earcons. Auditory icons consist of recorded non-speech sounds that can be recognized from real-world experiences. For instance, the sound of crumpling paper playing when deleting a digital document [2]. Earcons, on the other hand, are "audio messages used in the user-computer interface to provide information and feedback to the user about computer entities" [3]. Onboard an aircraft, short, structured musical messages can be used as alarms when a passenger requests assistance.

Auditory icons and earcons are instrumental in sound design across mediums like film, games, and human-machine interfaces, effectively conveying specific information to users. In gaming scenarios, for instance, they play a vital role in alerting players to significant events [3–6]. To fully harness the power of non-speech auditory cues, and most notably the synthetically-designed earcons resulting from a greater domain of creative freedom and sonic signifiers, designers must align these cues with their intended meanings. Despite a common understanding and shared recognition of the structural attributes inherent to each semantic adjective among designers across different cultures, there are no established guidelines or systematic analyses of their component structures.

Understanding the sonic attributes inherent to the earcon's design can shed light on human perception, psychology, and communication science by means of cross-modal metaphors, but also in guiding the design principles for optimal communication with sound. For example, the problem of creating an alert sound to signal errors must have a sonic identity that intuitively communicates the information to users.

Research into the structural attributes of semantic adjectives has predominantly utilized music perception methods, focusing on identifying significant attributes through perceptual listening tests. Investigations into the relation between sonic descriptors and the perception of musical instrument sounds have been conducted by various researchers. Disley and Howard [7] explored the timbres of pipe organs, while Bernays and Traube [8] delved into piano tones. Rosi et al. [9] analyzed the meaning of metaphorical sound attributes based on interviews with sound professionals. These studies, despite their differ-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ences, identified common descriptors such as "bright", "warm", and "round", highlighting consistent perception signifiers of musical timbres. A different line of inquiry has examined the impact of audio descriptors on semantic annotations. Studies have shown that the brightness of a sound is associated with its spectral centroid, attack time, and timbral sharpness. Following this approach, Ilkowska and Miskiewicz [10] delved into the effects of spectral variations on the perception of sound. By introducing artificial formants to both noise and music spectra and applying psychophysical scaling to assess their relationship, their research demonstrates that vowel frequency plays a pivotal role in altering perceptions of sharpness and brightness. This influence surpasses that of vowel bandwidth and amplitude, underscoring the critical impact of frequency on the auditory perception of semantic qualities.

Concerning the analysis of earcon-specific attributes, there is a lack of literature on assessing their intrinsic structure. To this end, we present an empirical analysis of bipolar [1] semantic adjectives in earcon design, as a first step towards an in-depth (perceptual) study to unveil the foundational sonic attribute of semantic adjectives. In detail, we infer common attributes from a newly complied dataset of earcons from *weak* user-driven tags, named BiSAID. It includes 599 sound samples for bipolar semantic adjectives covering 10 metaphors in different domains, e.g., temperature (cold vs. warm), and brightness (bright vs. dark). Samples were collected from the crowd-sourced Freesound platform. Samples were annotated with 66 instantaneous and global audio descriptors capturing spectral, temporal, spectrotemporal, and energetic information.

To evaluate the quality of compiled data and initiate the exploration of the sonic attributes of earcon design, we employ statistical and machine learning methods to deduce instantaneous earcon descriptors that effectively differentiate between bipolar adjectives within specific metaphorical domains. Our hypothesis posits that characteristic sonic dimensions, such as pitch, onset density, and spectral bandwidth, among others, are inherent to the sonic structure of earcons corresponding to various metaphors.

We present two detailed case studies. Firstly, we examine which descriptors in the BiSAID dataset are most pertinent for distinguishing between bipolar adjectives such as bright versus dark, and correct versus incorrect. The former represents a well-established baseline case, extensively discussed in existing literature, which should unveil the adequacy of our data in aligning with perceptual results and enlighten the limitations of the current sound collection. The latter introduces a novel metaphor that has received limited attention thus far. Treating each adjective as a distinct sample or class, we statistically infer significant differences between the two sample sets and determine their relative importance through machine learning models.

The reminder of the paper is structured as follows. Section 2 describes the BiSAID dataset structure and contents.

[1] This type of methodology on sound perception was first introduced by von Bismarck [11], based on the semantic differential technique, pioneered by Osgood [12], a method for measuring the meaning of concepts by eliciting ratings on bipolar adjective scales.

Table 1. Dataset composition: earcons, semantic labels, and categories.

Dimension	Adjectives	Category				#
		Earcon	Icon	Other	Exc.[a]	
Accuracy	correct	125	8	19	248	400
	incorrect	48	13	8	44	113
Brightness	bright	49	0	0	236	285
	dark	47	1	5	202	255
Shape	curved	43	0	30	327	400
	flat	27	0	10	308	345
Sharpness	sharp	46	54	69	231	400
	dull	14	49	32	202	297
Temperature	warm	55	0	59	286	400
	cold	0	20	82	298	400
		454	145	314	2382	3295

[a] Excluded due to erroneous tag.

Section 3.1 provides the description of two case studies on earcon semantic-driven descriptors for brightness (bright versus dark) and accuracy (correct versus incorrect). Section 4 presents preliminary results of the earcon data analysis and discusses the potential and limitation of our data. Finally, Section 5 presents the conclusions of our work and directions for future research.

2. BISAID: A DATASET OF EARCONS AND AUDITORY ICONS WITH SEMANTIC DIFFERENTIAL TAGGING AND ACOUSTIC DESCRIPTION

The BiSAID dataset has been meticulously classified and tagged to encompass the following semantic dimensions: temperature (cold vs. warm), brightness (bright vs. dark), sharpness (sharp vs. dull), shape (curved vs. flat), and accuracy (correct vs. incorrect). Each data point is annotated with semantic differential tags and described through a rich set of acoustic and psychoacoustic descriptors. These earcons and tags are summarized in Table 1, while the descriptors, comprising both global and instantaneous types, are detailed in the remaining of the current section.

The earcons within BiSAID were curated from Freesound [2] [13]. We aimed to collect up to 400 sounds for each tag, classifying them into three categories: *Earcons*, which are short musical sounds designed to convey specific information; *Auditory icons*, brief sounds that mimic familiar non-speech sounds from everyday life; and *Other*, which includes sounds relevant to the tag but not categorized as earcons or auditory icons. In the data cleansing phase, 2382 sounds were excluded due to inaccurate tagging. This step involved listening to each sound to assess if the tags accurately reflected the content, in terms of perception of sound. Despite the crowd-sourced nature of Freesound's tagging system and the resulting variability [14], the objective was to ensure the integrity of tags rather than to standardize their meanings. For example, a sound (id:574006) tagged as "correct" that

[2] `https://freesound.org`, last accessed on 27 February, 2024.

was unrelated to the concept of correctness was removed, as well as a sound (id:195219) tagged as "bright" which descriptors a male voice singing *Monty Python's* "Always Look on the Bright Side of Life"—a clear mismatch with the expected perception of brightness. The final dataset includes 3295 sounds, with 599 specifically identified as earcons or auditory icons.

For the computational analysis of earcons in the BiSAID dataset, we employed the Freesound API extractor, grounded in `Essentia` [15]. Initially, this tool computes a foundational set of 91 [3] primary low- and mid-level descriptors, encompassing a broad spectrum of acoustic and psychoacoustic properties. These foundational descriptors capture diverse acoustic dimensions, including temporal aspects (e.g., zero-crossing rate, effective duration), spectral characteristics (e.g., spectral centroid, spectral spread), spectro-temporal descriptors (e.g., spectral flux, the first derivative of spectral contrast), rhythmic elements (e.g., beats count, onset rate), and tonal attributes (e.g., tuning frequency, chord histogram).

Regarding the temporal analysis scale, we can group these descriptors in global descriptors, which synthesize characteristics over the entire sound duration (e.g., log-attack time, tristimulus), and instantaneous descriptors, calculated for discrete frames within the sound to provide a momentary acoustic snapshot (e.g., spectral rms, pitch salience). Building upon the initial set, the Freesound API further computes statistical aggregators for all instantaneous descriptors, thus expanding the dataset to encompass a total of 452 descriptors. These aggregators include the maximum, minimum, mean, variance, median, and the first and second derivatives, significantly enriching the dataset's descriptive capacity while capturing the acoustic temporal profile of each sound. Notably, approximately 20% of these descriptors are vector-based structures, such as the Equivalent Rectangular Bandwidth (ERB) [16, 17] bands or the Gammatone Frequency Cepstral Coefficients (GFCC), offering multidimensional insights into the sound's acoustic and psychoacoustic profile.

BiSAID is made available in several formats to cater to both manual and computational exploration. This encompasses Excel (`xlsx`) files, which aggregate all sound-related information by dimension, and CSV and JSON files that present audio descriptors alongside semantic labels. The content of these files is curated to include the sound's `id`, `name`, `sound` (a hyperlink for previewing the sound), and `tags` (the full list of tags associated with the sound). After an auditive inspection, additional annotations are made to document the `category` of the sound and any pertinent `note` that offers insight into its categorization. The dataset, distributed under a Creative Commons license, is accessible at `https://figshare.com/articles/media/BISAID/25377589`. To promote reproducibility and transparency, we further provide the scripts utilized in the data collection and analysis phases, ensuring that researchers can readily replicate or extend the work presented herein. The scripts can be found in the paper repository at `https://github.com/ZijingCaoo/sound_analysis`.

[3] For a detailed description of these descriptors, please consult `https://freesound.org/docs/api/analysis_docs.html`.

3. A STUDY ON THE RELATIONSHIP BETWEEN SOUND PARAMETERS OF EARCON AND SEMANTIC DESCRIPTORS

This section presents a preliminary exploration of the BiSAID dataset, emphasizing the semantic dimensions of brightness (bright vs. dark) and accuracy (correct vs. incorrect). Our goal is to uncover the relationship between sound parameters and semantic descriptors, shedding light on the structural underpinnings and metaphorical semantic qualities of earcons. The focus on earcons, in preference to auditory icons, stems from their inherent design process. Unlike auditory icons, which are based on sampled sounds from the environment, earcons are synthesized creations, meticulously crafted by sound designers to convey specific semantic meanings. This deliberate design process allows for a deeper investigation into how certain audio descriptors are intentionally manipulated to represent different semantic labels, offering a more controlled environment to study the precise relationship between sound parameters and their perceived meanings. Moreover, this investigation evaluates the dataset's quality and the data collection process, setting the groundwork for future research and the next phases of our study.

The choice of brightness and accuracy for our study is twofold: the well-documented discussion of brightness in music perception literature [7, 18, 19], and the relatively uncharted territory of the accuracy dimension, critical in sound design for indicating correct or incorrect actions [20, 21].

3.1 Method

To address our objectives, we began with two preliminary steps regarding earcon and descriptor selection. To prevent imbalances, we ensured an equal representation of earcons for each tag, selecting 47 earcons per tag, based on the maximum common count across descriptors for both pairs of correct / incorrect and bright / dark. In terms of descriptors, we aimed for a manageable and interpretable descriptor space, focusing on one-dimensional numerical descriptors and excluding vector-based and non-numeric attributes. Only the mean values of instantaneous descriptors were retained, setting aside other statistical aggregates to streamline our analysis. Additionally, we removed descriptors with constant values across all earcons, as they do not aid in our analysis. An example is the second-peak-spread (`rhythm.second_peak_spread.mean`, according to Freesound original naming) descriptor in the accuracy case study, which showed no variability and was thus excluded.

After establishing the groundwork, our analysis proceeds with three key stages aimed at clarifying how earcon parameters relate to semantic descriptors: statistical analysis of descriptors, descriptor importance ranking, and correlation analysis.

First, we assessed the normality of the distribution of these descriptors using the Shapiro-Wilk test [22]. The

test indicated that approximately 80% of the descriptors deviated from a normal distribution (more precisely, 84% for the accuracy case, and 74% for the brightness case), thus requiring the use of non-parametric tests for further statistical analysis. Consequently, we employed the Mann-Whitney U test [23] [4] to evaluate the statistical significance of differences between groups defined by semantic labels, ensuring the robustness of the descriptor selection process against non-normal distribution patterns. We used a p-value threshold of 0.001 to ensure that the observed differences in medians between the two groups were highly unlikely to have occurred by random chance.

The second phase of our analysis involved the use of two machine learning models, *logistic regression* and *random forests*, to rank the descriptors in terms of the predictive power of the semantic labels. This approach under two alternative machine learning models allowed the identification of both linear and non-linear relationships between the descriptors and the labels, and gave us an idea of the more important descriptors.

The convergence of statistical significance, and model-based predictive power led to a focused subset of descriptors for deeper analysis.

4. SOUND ANALYSIS AND RESULTS

In this section, we examine the correlations between earcon descriptors and semantic labels within two semantic dimensions: *accuracy* (contrasting "correct" vs. "incorrect" bipolar labels) and *brightness* (contrasting "bright" vs. "dark"). Each dimension's analysis follows a structured approach: initially presenting descriptor importance as determined by our computational models, with particular emphasis on descriptors that are statistically significant and hold high importance in both models. Subsequently, we analyze the distributions of these descriptors, providing visual comparisons to elucidate how they vary between semantic contrasts.

4.1 Case I: *Brightness*

Figure 1 presents the ranking of descriptors according to their predictive power for semantic earcon labels of brightness, differentiating between "bright" and "dark" labels. Descriptors that showed statistical significance, as determined by the Mann-Whitney U test with $p < 0.001$, are highlighted with an asterisk next to their labels.

The analysis reveals a notable alignment between the most statistically significant audio descriptors and the random forest method, suggesting the data's potentially non-linear characteristics. However, our focus remains on those descriptors that not only show high statistical significance but also are crucial in the results produced by both predictive models (random forest and logistic regression). Identified as highly relevant in both analyses and marked in dark blue in Figure 1, these descriptors predominantly pertain to the spectral domain

[4] A non-parametric method for comparing differences between two independent groups in instances where the dependent variable is ordinal or continuous yet does not adhere to a normal distribution.

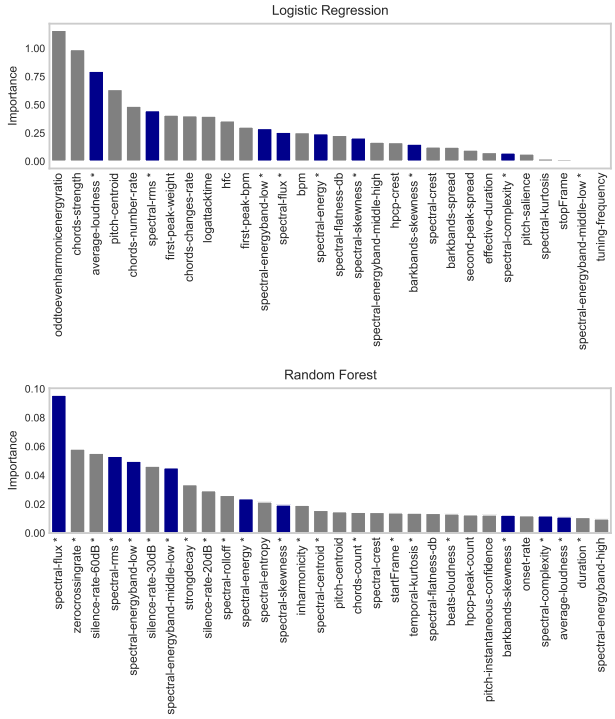


Figure 1. Descriptor importance for semantic label prediction for the *brightness* dimension. Asterisks (*) denote statistically significant descriptors; (*grey*) descriptors highlight high import descriptors in the corresponding model; and (*blue*) descriptors high importance in both models.

(such as *spectral-rms*, *spectral-energyband-low*, *spectral-energyband-middle-low*, *spectral-skewness*, *barkbands-skewness*, *spectral-complexity*, and *spectral-flux*), with the exception of *average-loudness*, which operates in the temporal domain. A significant number of these descriptors are associated with the sound's energy.

Due to the absence of energy normalization in our methodology, we refrain from drawing definitive conclusions about certain descriptors (*spectral-rms*, *spectral-energyband-low*, *spectral-energyband-middle-low*, *spectral-energy*, *average-loudness*). It is noted that "dark" distributions consistently exhibit higher values than "bright" ones in aspects like *average-loudness* and *spectral energy*, particularly at the lower end of the spectrum (*spectral-energyband-low*). This observation aligns with the intuitive association of "darker" sounds, in terms of both luminescence and mood (e.g., "scary"), with greater energy in the lower frequency range and increased loudness.

Given the high correlation between *spectral-skewness* and *barkbands-skewness* (differing only in the scale used for computation), we opt to exclude only the latter from further analysis, as shown in Figure 2, focusing on the former due to its direct relevance. *Spectral-skewness* quantifies the asymmetry of the spectrum's distribution around its mean, indicating more energy on the right-hand side of the distribution at lower values, and conversely, more energy on the left side suggests a higher skewness value.

This descriptor, therefore, is crucial for understanding the distribution of energy across the frequency spectrum.

In addition to *spectral-skewness*, other descriptors under consideration include *spectral-complexity*, which counts the number of peaks in the spectrum between 100Hz and 5KHz, and *spectral-flux*, which gauges the changes in the frequency content over time. The latter uses either the L2- or L1-norm difference between consecutive frames of the magnitude spectrum, offering insights into how the spectral content evolves. These descriptors collectively provide a multifaceted view of the sound's spectral characteristics, illuminating different aspects of how sounds are perceived as either "bright" or "dark".

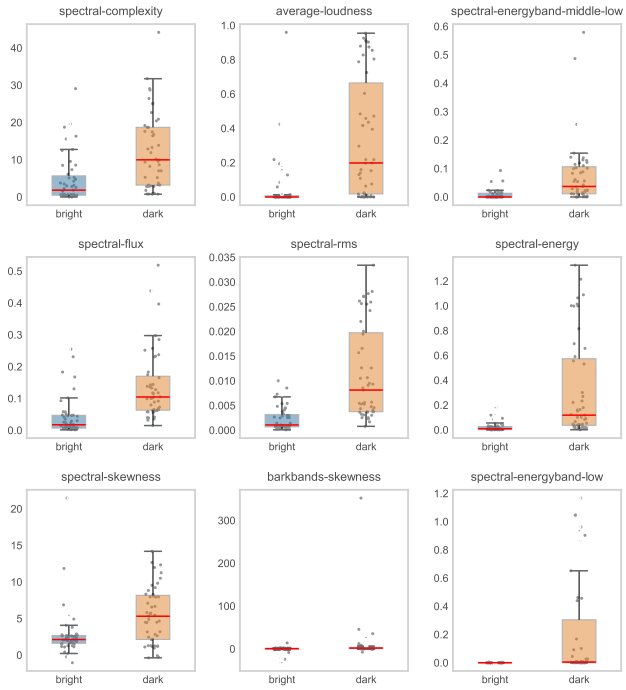


Figure 2. Distribution analysis of key acoustic descriptors for the dimension of *brightness*, contrasting "bright" vs. "dark" semantic labels.

Attempting to interpret the data, the higher skewness values in dark earcons compared to bright ones suggest a trend where the spectral content of "dark" sounds is skewed towards lower frequencies. The broader range of mean values for "dark" instances may imply less consistency in the spectral centroid, hinting at an intuitive connection where "dark" sounds are associated with lower-frequency energy. This observation, while not conclusive, resonates with a common perception that links auditory darkness with a prevalence of lower frequencies.

In the analysis of *spectral-complexity*, dark earcons show a greater complexity than their bright counterparts, particularly in sounds tagged as "scary" or "creepy" on Freesound. This increased complexity might intuitively suggest that "dark" sounds encompass more intricate spectral textures and a higher level of dissonance from overlapping spectral peaks. While this insight does not serve as a definitive explanation, it opens up a pathway for intuitive speculation on how complexity in the spectral domain could influence our perception of sounds as being "dark" or ominous.

Regarding *spectral-flux*, the observation that dark earcons exhibit a higher flux raises speculative thoughts on the nature of these sounds. The higher spectral flux indicates more variability in the spectrum over time, which might intuitively be linked to the presence of elements such as low-frequency rumbles, atonal textures, and noise. This characteristic of "dark" sounds, suggesting a more dynamic spectral evolution, allows for speculative reflection rather than firm conclusions. It hints at how such auditory features could be perceived as "dark" aligning with a common intuition about the complexity and variability in sounds labeled as such.

4.2 Case II: *Accuracy*

Figure 3 delineates the hierarchy of descriptors based on their efficacy in predicting semantic earcon labels concerning accuracy, particularly distinguishing between "correct" and "incorrect" labels. Descriptors achieving statistical significance, as verified through the Mann-Whitney U test with $p < 0.001$, are distinguished by an asterisk beside their names.

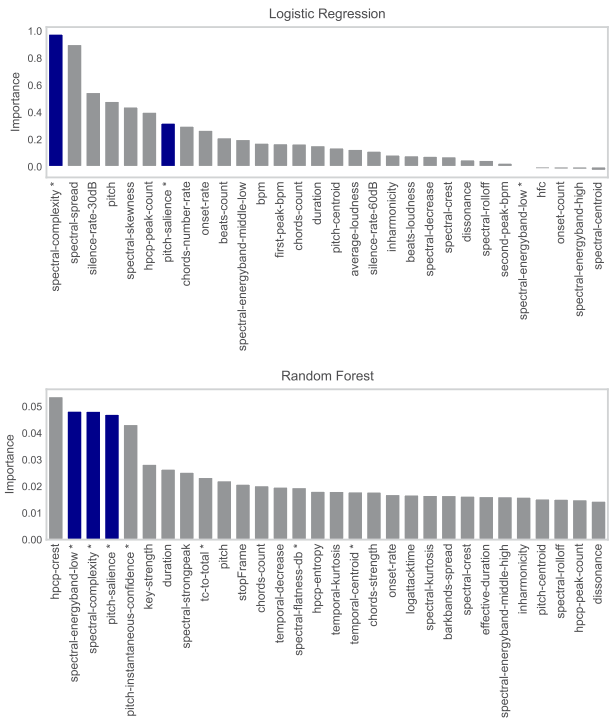


Figure 3. Descriptor importance for semantic label prediction for the *accuracy* dimension. Asterisks (*) denote statistically significant descriptors; (*grey*) descriptors highlight high import descriptors in the corresponding model; and (*blue*) descriptors high importance in both models.

Mirroring the analytical approach of the previous case, we concentrate on the intersection of highly significant descriptors as identified by both logistic regression and

random forest models, illustrated in dark blue in Figure 3. From this convergence, three audio descriptors stand out: *spectral-complexity*, *pitch-salience*, and *spectral-energyband-low*. Descriptive statistics for these salient descriptors are depicted in Figure **??**. Preliminary scrutiny of *spectral-energyband-low* indicates a negligible perceptual disparity amidst subtle variations.

The data distribution reveals an uptick in *spectral-complexity* for earcons labeled as incorrect relative to those deemed correct. Closer inspection of the earcons across our dataset exposed clear timbral distinctions between the two groups. Incorrect earcons typically embodied a multitude of percussive sounds characterized by densely populated spectral bands, whereas correct earcons were more likely to manifest as pitched tones within harmonic spectral frameworks. Thus, we may infer that the "incorrect" earcons are typified by a more intricate spectral profile, with a higher density of spectral peaks and a less prominent fundamental frequency.

This narrative of *spectral-complexity* is corroborated by the pitch salience assessments; incorrect earcons are marked by higher *pitch-salience* values. *Pitch-salience* is calculated as the quotient of the spectrum's highest autocorrelation peak over the unshifted autocorrelation baseline, reflecting the prominence of a tone. Sounds defined by pure tones or limited harmonics tend to register lower *pitch-salience* readings, tending towards zero, as noted in correct earcons. In contrast, the presence of multiple harmonics in the spectrum of incorrect earcons results in increased *pitch-salience*, indicative of their intricate harmonic structures.

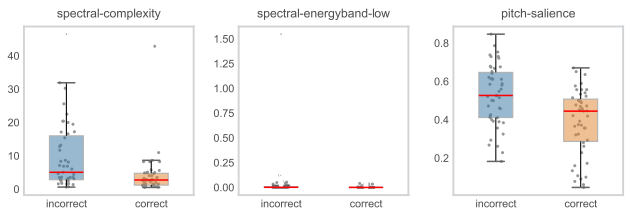


Figure 4. Distribution analysis of key acoustic descriptors for the dimension of *accuracy*, contrasting "correct" vs. "incorrect" semantic labels.

4.3 Discussion

This exploratory study has highlighted several key points regarding the use of audio descriptors for semantic description of sound.

The examination of audio descriptors in relation to semantic bipolar concepts such as "bright" versus "dark" and "correct" versus "incorrect" has demonstrated the potential of descriptors like *spectral-complexity* and *spectral-skewness* to reflect auditory qualities ascribed to these terms. However, the application of these descriptors is not straightforward due to the subjective nature of sound perception. The use of the descriptor "dark" raises issues concerning the alignment between user-generated tags and the a shared perception they are intended to characterize. Literature suggests that subjective tagging can introduce inconsistencies [14], that require further methodological treatment. Bipolar scales can compound this problem by presenting a false dichotomy that may not mirror the auditory experience, as discussed by Kendall and Carterette [18]. An alternative approach using unipolar scales, like the VAME method, avoids such binary oppositions by considering attributes on a continuum, which might better represent the spectrum of auditory qualities.

While the use of crowd-sourced tags offers controlled semantic judgment, it also requires a cautious approach due to its inherent limitations [24]. This necessitates more rigorous data collection methods to ensure the validity of earcon descriptors used in research.

In the context of accuracy, descriptors that might typically indicate sound correctness, such as dissonance or roughness, were not prominent in the dataset, suggesting a gap that warrants further exploration. However, the prominence of *pitch-salience* and *spectral-complexity* could be indirectly linked to these concepts, meriting a future intercorrelation analysis.

The dataset's descriptors showed great deviation from a normal distribution, indicating a complex set of factors at play in computational-based sound perception studies. These include the properties of the sounds or the recording conditions, to mention a few. Such atypical distributions in the descriptors necessitate meticulous outlier management and data transformation in preprocessing to ensure data quality, so not to introduce noise in subsequent analysis. This is especially true when using average-based temporal aggregation methods, which are more vulnerable to outlier effects than those based on median aggregation.

To refine the descriptor extraction process, employing tools like Essentia[5] is recommended. Essentia provides enhanced control over the analysis parameters, allowing for a more robust and controlled extraction process. Utilizing such a tool could improve the precision of sound data analysis, leading to more reliable and significant research findings.

5. CONCLUSIONS AND FUTURE WORK

This study introduces a dataset comprising 1053 non-speech auditory cues, including earcons and auditory icons, each annotated with semantic bipolar adjectives and detailed through 452 global and instantaneous audio descriptors. Our focus was on analyzing a subset of 192 earcons, labeled with contrasting pairs "correct" and "incorrect," and "bright" and "dark," to examine the relationship between earcon parameters and their semantic meanings.

The analysis of semantic earcon labels for brightness differentiation unveils important audio descriptors contributing to perceptual differentiation. Through logistic regression and random forest methods, statistically significant descriptors such as energy-related descriptors, spectral skewness, complexity, and flux are identified. Notably, brighter earcons exhibit lower energy levels, while

[5] https://essentia.upf.edu/

darker ones display higher positive skewness, reflecting timbral characteristics akin to pitched instruments. Spectral complexity analysis reveals intricate frequency components in darker earcons, evident in dense spectral patterns and sensory dissonance. Additionally, spectral flux analysis demonstrates temporal variations, with darker earcons showing greater spectral content variability. These findings offer insights into the nuanced perceptual attributes of earcons, informing future auditory design considerations.

From the analysis of the accuracy study case, featuring correct and incorrect earcons and similar methodology, three key audio descriptors have been highlighted: spectral complexity, pitch salience, and spectral energy band low. Descriptive statistics for these descriptors uncovered higher spectral complexity in incorrect earcons, characterized by dense percussive sounds, in contrast to correct earcons featuring harmonic tones. This finding was reinforced by pitch salience analysis, which indicated variations in harmonic content between correct and incorrect earcons.

Looking ahead, our research paves the way for several avenues in future work. We anticipate the integration of additional models or ensemble methods, such as Gradient Boosting Machines (GBM) [25], to further substantiate and refine the descriptor importance rankings, particularly for capturing complex non-linear relationships. The expansion of our dataset, incorporating a wider variety of sounds selected through a more stringent methodology, stands as a priority. This will enable us to extend our analysis to additional dimensions of earcon design, ultimately extracting actionable insights for sound designers.

Moreover, the true test of our findings lies in their perceptual validation. We propose a perceptual study to confirm the predictive power of the identified descriptors in real-world settings, ensuring that our statistical and machine learning outcomes resonate with actual perceptual differences among listeners. Complementary to this, controlled listening tests with participants will allow us to assess their perception of brightness and accuracy in earcons, aligning subjective assessments with the quantitative data from our analysis.

Through this interconnected approach, we aim to bridge the gap between the theoretical underpinnings of earcon parameters and their practical design applications, fostering a deeper understanding and more intuitive design of auditory signals.

6. ACKNOWLEDGEMENT

The first author would like to thank China Scholarship Council (CSC)[6] for financial support (Grant No.202307920001).

7. REFERENCES

[1] T. Görne, "The emotional impact of sound: A short theory of film sound design," in *EPiC Series in Technology*, 2019, pp. 17–30.

[2] T. Hermann, A. Hunt, and J. G. Neuhoff, *The sonification handbook*. Logos Verlag Berlin, 2011.

[3] M. M. Blattner, D. A. Sumikawa, and R. M. Greenberg, "Earcons and icons: Their structure and common design principles," *Human-Computer Interaction*, vol. 4, no. 1, pp. 11–44, 1989.

[4] R. D. Patterson, J. Edworthy, and M. C. Lower, *Alarm sounds for medical equipment in intensive care areas and operating theatres*. University of Southampton, Institute of Sound and Vibration Research, 1986.

[5] S. Brewster, V.-P. Raty, and A. Kortekangas, "Earcons as a Method of Providing Navigational Cues in a Menu Hierarchy," *People and Computers XI*, no. August, pp. 169–183, 1996.

[6] G. Leplâtre and S. A. Brewster, "Designing Non-Speech Sounds to Support Navigation in Mobile Phone Menus," in *Proceedings of the International Conference on Auditory Display (ICAD)*. University of Glasgow (United Kingdom), 2000, pp. 190–199.

[7] A. C. Disley and D. M. Howard, "Spectral correlates of timbral semantics relating to the pipe organ," *Speech, Music and Hearing*, vol. 46, pp. 25–39, 2004.

[8] M. Bernays and C. Traube, "Verbal expression of piano timbre: Multidimensional semantic space of adjectival descriptors," in *International Symposium on Performance Science*, no. July, 2011, pp. 299–304.

[9] V. Rosi, O. Houix, N. Misdariis, and P. Susini, "Investigating the Shared Meaning of Metaphorical Sound Attributes," *Music Perception*, vol. 39, no. 5, pp. 468–483, 2022.

[10] M. Ilkowska and A. Miśkiewicz, "Sharpness versus brightness: A comparison of magnitude estimates," *Acta Acustica united with Acustica*, vol. 92, no. 5, pp. 812–819, 2006.

[11] G. von Bismarck, "Timbre of Steady Sounds: A Factorial Investigation of its Verbal Attributes," *Acustica*, vol. 30, pp. 146–159, 1974.

[12] C. E. Osgood, G. J. Suci, and Percy H. Tannenbaum, *The Measurement of Meaning*. University of Illinois Press, 1957.

[13] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *MM 2013 - Proceedings of the 2013 ACM Multimedia Conference*, 2013, pp. 411–412.

[14] F. Font and X. Serra, "Analysis of the folksonomy of Freesound," in *Proceedings of the CompMusic Workshop*, 2012, pp. 48–54.

[15] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: An Audio Analysis Library for Music Information Retrieval," in *International Society for Music Information Retrieval (ISMIR)*, 2013, pp. 2–7.

[6] https: //www.csc.edu.cn.

[16] B. C. J. Moore and B. R. Glasberg, "Suggested formulae for calculating auditory-filter bandwidths and excitation patterns," *The journal of the acoustical society of America*, vol. 74, no. 3, pp. 750–753, 1983.

[17] B. R. Glasberg and B. C. J. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing research*, vol. 47, no. 1-2, pp. 103–138, 1990.

[18] R. Kendall and E. Carterette, "Verbal attributes of simultaneous wind instrument timbres: I. von bismarck's adjectives." *Music Perception*, vol. 10, no. 4, pp. 445–467, 1993.

[19] S. McAdams, B. Giordano, P. Susini, G. Peeters, and V. Rioux, "A meta-analysis of acoustic correlates of timbre dimensions," *The Journal of the Acoustical Society of America*, vol. 120, no. 5_Supplement, pp. 3275–3276, 2006.

[20] E. Rovithis, A. Floros, N. Moustakas, K. Vogklis, and L. Kotsira, "Bridging audio and augmented reality towards a new generation of serious audio-only games," *Electronic Journal of e-Learning*, vol. 17, no. 2, pp. 144–156, 2019.

[21] S. K. Adams and L. B. Trucks, "A procedure for evaluating auditory warning signals," in *Proceedings of the Human Factors Society Annual Meeting*, vol. 20, no. 8. SAGE Publications Sage CA: Los Angeles, CA, 1976, pp. 166–172.

[22] S. S. Shapiro and M. B. Wilk, "An analysis of variance test for normality (complete samples)," *Biometrika*, vol. 52, no. 3-4, pp. 591–611, 1965.

[23] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947.

[24] A. C. Disley, D. M. Howard, and A. D. Hunt, "Timbral description of musical instruments," in *International Conference on Music Perception and Cognition*, 2006, pp. 61–68.

[25] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of statistics*, pp. 1189–1232, 2001.

"BACK IN MY DAY...": A PRELIMINARY STUDY ON THE DIFFERENCES IN GENERATIONAL GROUPS PERCEPTION OF MUSICALLY-EVOKED EMOTION

Pedro L. LOURO (pedrolouro@dei.uc.pt) (0000-0003-3201-6990) [1] and
Renato PANDA (panda@dei.uc.pt) (0000-0003-2539-5590) [1,2]

[1] *CISUC, DEI*, **University of Coimbra**, Coimbra, LASI, Portugal
[2] *Ci2 — Smart Cities Research Center*, **Polytechnic Institute of Tomar**, Tomar, Portugal

ABSTRACT

The increasingly globalized world we live in today and the wide availability of music at our fingertips have led to more diverse musical tastes within younger generations than in older generations. Moreover, these disparities are still not well understood, and the extent to which they affect listeners' preferences and perception of music. Focusing on the latter, this study explores the differences in emotional perception of music between the Millennials and Gen Z generations. Interviews were conducted with six participants equally distributed between both generations by recording their listening experience and emotion perception on two previously compiled sets of songs representing each group. Significant differences between generations and possible contributing factors were found in the analysis of the conducted interviews. Findings point to differences in the perception of energy of songs with specific messages of suffering for love, as well as a tendency from the younger group to perceive a well-defined emotion in songs representing their generation in contrast to neutral responses from the other group. These findings are preliminary, and further studies are needed to understand their extent. Nevertheless, valuable insights can be extracted to improve music recommendation systems.

1. INTRODUCTION

In music psychology, many studies on the factors that contribute to the differences when reporting musically-evoked emotion show that these are related to background, genre, personality, musical education, taste, and repertoire [1]. These are normally irrelevant when dealing with perceived emotion, but evidence shows that cultural background may indeed influence emotion perception in music [2]. These studies are still relatively recent and normally focus on the differences between two distinct cultures. But what if we consider the evolution of cultures? Various studies show that certain generations have generalized music preferences, such as Baby Boomers (around 1946 through 1964), who have a Rock preference, or Generation X (around 1965 through 1980) who prefer Grunge and Hip-Hop, which reflect the views of society, with or against it, at each time period. Despite this, mainly due to the growing globalization of culture and the widespread access to music, preferences have become difficult to attribute to younger generations, such as Millennials and Generation Z (or Zoomers). Although for previous generations, we can deduce that music emotion perception has some similar bias transversal to each generation, this is not so clear for the more recent ones [3].

Two research questions were formulated in order to understand potential differences in perception across the more globalized generations: i) How different is the emotional perception between Millennials and Zoomers?; ii) What factors contribute to emotional perception differences? To answer the first question, the perceived emotion in a set of songs must be collected from various members of both generations. This can be done by showing a small excerpt of those songs and explicitly asking for their emotional content, not how it made them feel. To explain the previous answers and potential differences, the second question needs to be answered, which can be done by finding common themes between the various members' realities.

To answer these questions, a study is conducted following a phenomenological approach that first contextualizes the musical trends represented by music popular at the end of the generation into which the subjects fall. The subjects are asked to recount their experience with each song and their overall experience with the songs listened to. The same is conducted for the musical trends of the other generation. Therefore, the study adopts a social constructivism lens to take into account the multiple realities described by the different participants. Contributions from this study are twofold: First, evidence of differences in perception between both generational groups is provided across the two generation's musical trends. Second, possible factors contributing to such differences are described, such as the different perception of energy in songs that deal with the message of suffering for love and the younger generation's preposition to perceive more pronounced emotions in songs that convey low emotional intensity in contrast to the older generation.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Figure 1. Russell's Circumplex Model adapted from [4]. As seen in the figure, each quadrant represents an emotion, but should these values be low enough the emotion is considered neutral. At = Arousal threshold; Vt = Valence threshold.

2. BACKGROUND

2.1 Emotion and Music

From the rituals conducted by our far-off ancestors to adjusting our mood according to the task at hand in the present day, such as in a professional context, music has always been very important to our way of life. Inherent to thinking of music is the emotion that comes attached to it, regarded as the main reason why we engage with it [5] and which nature has been a debate for many centuries [6]. Some argue that the emotions evoked by music are merely perceived by the listener and not actually induced within them; others defend the opposite notion and the most accepted idea that both are present. However, induced emotions are prone to high subjectivity, which makes it hard to find agreement when asking subjects to report their experiences [7], and this is why perceived emotion is preferred in most contemporary quantitative studies as it is considerably less subjective.

Furthermore, there is also a need to represent the whole emotion spectrum to collect annotations for these studies, for which emotion models exist. These can be either categorical or dimensional. The former is usually divided into a set number of clusters or groups of related emotions, the most commonly used being Hevner's Adjective Circle [8], and is easy for annotators to understand. However, categorical models do not accurately represent the emotion spectrum due to the strict boundaries between clusters when some emotions may lie there. Dimensional models solve this problem by mapping emotions into a continuous space on several planes, such as the Russell's Circumplex Model [9] seen in Figure 1, which most commonly is represented by two planes related with human biological systems for differentiating emotions, these being arousal, the amount of energy present in the stimuli, and valence, how negative or positive this stimulus is.

Most literature regarding the fields that study emotion in music uses the latter model due to its accurate spectrum representation. However, it is known that annotators tend to have difficulty understanding the workings of the model and need some previous guidance to provide accurate annotations [10]. The wide adoption of this model can be seen, for example, in developing datasets for Music Emotion Recognition, such as the DEAM dataset [11].

2.2 Influences On Emotion Perception

One of the major concerns of music psychology when dealing with emotions is the set of factors that influence subjects' emotional responses. As most music is not usually confined to specific contexts, various individual factors have been considered, such as age, gender, music education, listening habits, and personality [1, 12, 13], to name a few.

Researchers have also extensively studied cultural background as a critical factor, emphasizing its impact through differences in traditions, societal norms, and diverse music theories across the globe. To some extent, some studies have found common emotion perceptions between Western and indigenous African cultures [14]. These are, however, limited to some of the basic emotions from Ekman's studies [15]. More thorough studies on Western and Eastern cultural differences in emotion perception showed significant discrepancies [2].

These studies focus on different cultures, and to the authors' knowledge, no study focuses on the differences between two generational groups with similar cultural backgrounds.

3. METHOD

3.1 Participants

The participants for this study were mainly chosen to fit into that particular generation, although different ages from the already interviewed participants were preferred later on. A subject is considered a Millennial or Zoomer between 43-26 and 25-11 years of age, respectively. We proceed to report the most relevant information from the participants.

The proportion of subjects per generation to which they belong was even, with three subjects in each. As for the distribution of ages within these generations, Millennials were more encompassing of that generation than Zoomers, as can be seen by the average age being 33, around the middle of that generation. However, for Zoomers, there is a higher concentration at the beginning since the age mean is around 23, close to the boundary previously defined. Although not considered relevant for this study, the proportion of the subject's gender is heavily male-skewed, which may lead to some bias on the conclusions taken from the individual experience. Participants are referred to as P*, where * is a number attributed to decreasing age, meaning that P1 refers to the oldest participant and P6 to the youngest.

As for music education, half of the subjects only received basic music education as part of their school curriculum (P3, P5, P6), while the other half have different levels of expertise, from basic music theory (P2) to instrument-specific (P1), to choir-specific education (P4). Finally, regarding music taste, there is a variety of preferred genres. When considering Millennials, there is little to no overlap

Table 1. Participants' individual information.

	Generation	Age	Gender	Professional Background	Previous Music Education	Frequency of Music Listening	Preferred Genres
P1	Millennials	42	M	Studying Computer Science; Worked on the same field	Self-learned: Intermediate (Applied to Guitar)	At least 3 days per week	Rock, Jazz, Blues, Reggae, Classic
P2		31	M	Studying Computer Science; Working on Artificial Inteligence	Self-learned: Basic	Every day	Samba, Bossa Nova, House
P3		27	M	Studying Computer Science; Studied Cinema	Basic School Education	Every day	Pop, EDM, K-Pop, Punk Rock, Trance, Classic
P4	Zoomers	25	M	Studied Museology; Working in a Scientific Center	Choir-educated	Every day	Pop, Rock, Latino, KPop
P5		24	F	Studying Languages and Literature	Basic School Education	At least 3 days per week	Pop, Indie Rock, Lo-Fi
P6		21	M	Studying Civil Engineering	Basic School Education	Almost every day	Heavy Metal, Metal, Rock

between subjects' preferences, while there is some overlap in Zoomers, specifically in Rock. This information is summarized in Table 1.

3.2 Material for Conducting the Study

For conducting the data collection process, a playlist representing the overall music trends for each generation was compiled. Properly defining what music trends needed to be represented for each generation would be another study altogether, so a more straightforward approach was followed.

First, a point in time representative of each group had to be defined. Achieving this is challenging because the goal is to capture most musical trends within each generation's defined 15-year period. The chosen approach was to establish reference points at the years that signify the start of the succeeding generation, i.e., 1997 for Millennials and 2012 for Zoomers. This last one was extrapolated from previous generations' trends as this point lacks general agreement.

Having defined these points, a set of songs from those years needed to be chosen. At this stage, it was decided to only account for songs described as Pop due to their mainstream appeal. The amount of songs that present conflicting emotional content, i.e., ambiguous emotion, or that lack emotion in general, is also considerably reduced compared to other genres. Songs were pulled from Billboard's Hot 100 chart from each point in time, aggregating the top 100 songs played in the USA each week since the 1960s. Billboard was chosen as the provider for this data as it represents the most influential region for this genre, the United States of America, for the relevant time periods.

With the songs picked, excerpts representative of those songs' emotional content were obtained to conduct the study. The 30-second previews on Spotify were chosen for this purpose. The choice was evident since this streaming platform has one of the biggest libraries in the market, decreasing the probability of no match being found from each chart. It also allows playback of the mentioned previews from a web browser at no charge for non-commercial use. There is, however, the caveat of there being no guarantee that these previews represent the song's emotional content. After reviewing them, the authors consider them adequate for the study, but this point should be further addressed in future studies. A script was developed to automatically create a Spotify playlist from the provided Billboard's Hot 100 chart week, where all previews' URLs were extracted and saved. The final playlists are shown in Table 2.

3.3 Data Collection

All interviews began by asking for relevant personal information from each participant, including age, gender, place of birth, and professional situation. This information is necessary to place the participant into one of the study's generations and understand possible bias behind their answers to the following questions. In addition, we also collected the level of musical education, frequency of music listening, and music preferences data for the same purpose. The collection process continues with the participant listening to the previously compiled music excerpts of six songs representative of their generation.

After each excerpt, three questions are asked:

- 'Do you have past experiences with this song?', to understand if there is any influence to their judgment on the emotional perception of that excerpt derived from past experiences;
- 'How would you describe your overall experience?', in an effort to separate the induced from the perceived emotion;
- 'What is the predominant emotion you perceive on this excerpt?'.

This query should output the perceived emotion from the song without influences from previous experiences and provide some insight into the factors behind it.

At the end of the playlist, the participant is questioned about their overall experience with the set of songs. They are encouraged to describe their differences and similarities, providing more insights into the emotion perceived in each song. The process is replicated for the playlist of the alternate generation. To conclude the interview, participants are asked to reflect on the two playlists, more specifically, identify recurring themes and highlight the most relevant differences.

3.4 Data Analysis

In order to better analyze the answers given for the perceived emotion in each song, each is mapped to Russell's Circumplex model, previously presented in Section 2. For answers that fall in one of the quadrants, we consider the overall emotion as the predominant emotion of that quadrant. For example, should responses perceive sadness and depression in one particular song, the overall perception would be sad since both reported emotions fall into Q3,

Table 2. The final playlists used in the study - one for each generation.

Millennials	Zoomers
"Semi-Charmed Life" by "Third Eye Blind"	"What Makes You Beautiful" by "One Direction"
"Bitch" by "Meredith Brooks"	"Set Fire to the Rain" by "Adele"
"Quit Playing Games (With My Heart)" by "Backstreet Boys"	"Payphone" by "Maroon 5"
"Un-Break My Heart" by "Toni Braxton"	"We Found Love" by "Rihanna"
"Say Youll Be There" by "Spice Girls"	"Call Me Maybe" by "Carly Rae Jepsen"
"All By Myself" by "Céline Dion"	"Titanium (feat. Sia)" by "David Guetta"

mainly related to sadness. In cases where the answers reveal uncertainty or conflicts between quadrants, the overall emotion is marked as neutral since it does not fit entirely into a particular quadrant.

The next step was to extract overall themes from the interviews that could explain differences between and within generations. The steps are outlined next, following the process described in Cresswell [16].

Each interview was transcribed from their respective recordings before being broken up into smaller meaningful units, i.e., evidences, for further analysis. The transcripts were read, and memoing was conducted in order to provide a more general view of the data and identify common themes between participants. Some concepts, i.e., codes, were identified by cross-referencing the relevant evidence in a spreadsheet.

The document organization comprehends the participant, its generation, and the corresponding playlist that produced the evidence. The emerging codes from the interviews followed these. When evidence is considered relevant, this is marked accordingly. Finally, correlations between each code were analyzed, grouping these codes into more general themes.

4. RESULTS

This section presents the findings of the present study. A thorough description of the differences in perception between both generations is given before presenting the common themes found in the interviews that may factor into their answers.

4.1 Overall Findings

For assessing the agreement between generations' emotional perception of each song, the mode of the answers was considered as the overall answer for each generation. Of the twelve songs used for this study, only five had full agreement. The emotion accounts discussed below refer to perception only if not stated otherwise. We consider an account neutral only if no other emotion is perceived by the participant at any point.

Regarding the Millennials playlist, three out of the six songs do not show agreement, namely "Bitch" by "Meredith Brooks", "Say You'll Be There" by "Spice Girls", and "Quit Playing Games (With My Heart)" by "Backstreet Boys". Participants from the Zoomer generation perceived all of these songs as Relaxed, while Millennial participants did not perceive any concrete emotion. In other words, the emotional content was perceived as neutral in most of the songs, except the last one mentioned above, which was perceived as happy.

Interestingly, there was even less agreement considering the Zoomers' set of songs, from which four out of six songs did not receive agreement between the generations, namely "Payphone" by "Maroon 5", "We Found Love" by "Rihanna", "Titanium" by "David Guetta feat. Sia", and "Set Fire to the Rain" by "Adele". Starting with "Payphone", most of the Zoomer generation considered this song sad, while Millenians reported more neutral responses. The latter also reported neutral responses for "We Found Love" and "Titanium", while the former perceived these songs as happy. As for "Set Fire to the Rain", Zoomers mostly perceived the song as sad, although one participant from this group perceived it as angry. Millennials mostly perceived it as angry, except for one participant who perceived a happier emotion.

From the accounts regarding their own playlists, it was clear that the older generation is more attached to the provided set of songs. When explicitly asked if they identified themselves with their generation's playlist, Zoomers tended to discuss how it did not match their taste. This finding corroborates the more sparse tastes in younger generations as discussed back in Section 2.

4.2 Factors Contributing to Emotion Perception

From the data analysis of the interviews described in Section 3.4, various common themes were found that may not only explain the observed differences, but also explain the large number of neutral responses mentioned above. In addition, some themes also show that the individual experience does not influence the emotion perception of the participants. Descriptions of codes and themes discussed in this section can be found in Table 4.

4.2.1 Past Experiences Do Not Influence Judgement

Many accounts show some influence of nostalgia on the recounts of the listening experience from each participant. Sometimes very explicitly, ''*A mix of nostalgia and happiness.*'(P5, "Payphone"), ''*(The song) gives me nostalgia and calm (feelings)...*'(P4," Say You'll Be There"), ''*Nostalgia, simpler times...*'(P6, "Semi-Charmed Life"). Many times these experiences are bound to some specific context, such as time with specific periods of life, '*Something an angsty teenager would hear.*'(P2, "Semi-Charmed Life"), a social context, '*I associate ("We Found Love") a lot with going out at night as a teenager.'(P3)*, or even social behavior, '*... we want to dance this with that girl we like*'

Table 3. Emotions perceived by each subject for both playlists.

Generation	Song	P1	P2	P3	P4	P5	P6	Overall Millennials	Overall Zoomers
Millennials	"Semi-Charmed Life" by "Third Eye Blind"	Sad	Happy	Happy	Happy	Sad	Happy	Happy	Happy
	"Bitch" by "Meredith Brooks"	Neutral	Neutral	Neutral	Relaxed	Happy	Relaxed	Neutral	Relaxed
	"Quit Playing Games (With My Heart)" by "Backstreet Boys"	Happy	Happy	Happy	Happy	Relaxed	Relaxed	Happy*	Relaxed*
	"Un-Break My Heart" by "Toni Braxton"	Sad	Sad	Sad	Sad	Sad	Sad	Sad	Sad
	"Say You'll Be There" by "Spice Girls"	Neutral	Neutral	Happy	Happy	Relaxed	Relaxed	Neutral	Relaxed
	"All By Myself" by "Céline Dion"	Sad	Angry	Sad	Sad	Sad	Sad	Sad	Sad
Zoomers	"What Makes You Beautiful" by "One Direction"	Happy	Happy	Happy	Happy	Happy	Happy	Happy	Happy
	"Set Fire to the Rain" by "Adele"	Neutral	Angry	Neutral	Sad	Angry	Sad	Angry	Sad
	"Payphone" by "Maroon 5"	Neutral	Neutral	Neutral	Sad	Neutral	Sad	Neutral	Sad
	"We Found Love" by "Rihanna"	Sad	Neutral	Neutral	Happy	Neutral	Happy	Neutral	Happy*
	"Call Me Maybe" by "Carly Rae Jepsen"	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral	Neutral
	"Titanium (feat. Sia)" by "David Guetta"	Sad	Neutral	Neutral	Neutral	Angry	Happy	Neutral	Happy**

*Emotions reported were of low intensity, but not neutral.
**Overall disagreement within generation; Extra comments from the participants were considered to map overall emotion.

(P1, "Un-Break My Heart"). Another common theme, especially in the set of songs for the younger generation, was the effect of continuous exposure to a given song that heavily impacted the listening experience. The exposure could either lead to aversion, '*Frustration, given the times I heard this song before.*'(P2, "What Makes You Beautiful"), or indifference, '*Plays a lot in the radio... it feels very neutral to me.*'(P1, "Payphone"). Even though there are some biases regarding the listening experience, this did not translate to the emotion perceived in the song, as seen in Table 3. This theme does not provide arguments for the differences between generations but confirms that perceived and induced emotion accounts were adequately separated.

4.2.2 Disagreeing On Suffering For Love

An extensively studied topic in music psychology is the paradox of negative emotions. The paradox refers to listeners reporting pleasurable experiences from music pieces in which emotional content is perceived as negative, e.g., '*Happy to hear this song (describing listening experience)... Depressive, to the point of cutting my wrists (describing perceived emotion)*'(P2, "Un-Break My Heart"). Such pleasurable experience may be explained by overwhelming vocals, namely "Un-Break My Heart", "All By Myself" and "Set Fire To The Rain". However, the underlying message of suffering for love seems to be the most important factor in songs perceived differently in the Millennial's playlist between generations. Many accounts seem to agree that most of that generation's music focuses on searching for and suffering after losing 'the one'. This translated to a happy or neutral perception for Millennials, while for Zoomers, it translated to perceiving relaxed emotions.

4.2.3 Disagreeing on Low-Intensity Emotions

Most of the oldest generation tends to respond neutrally to Zoomers' playlist songs, with four out of six songs perceived this way by the majority. In contrast, only "Call Me Maybe" received a similar neutral perception among the younger generation. A more concrete example is the accounts from Millennials about the song "Payphone". These reveal confusion derived from the disconnect between the instrumental and vocal emotional content, '*(Regarding my listening experience) I am very confused... The instrumental and vocals are very contradictory*'(P2, "Payphone"). The accounts in Table 3 reflect this in lower perceived intensity, as all accounts were Neutral across the generation. Similar to the previous theme, the main factor is the underlying message of self-empowerment identified by participants in both generations. One of the accounts nicely sums up this point: '*It is always the same thing, nostalgia, depression and there is no way out of this (referring to the suffering for love message of the Millennials' set of songs). But in 2012 (Zoomers' playlist), it is completely different because there is also depression, but things can turn around. There is more than just searching for love*' (P5). Overall, Millennials always perceive low-intensity emotions as Neutral in the Zoomer's playlist regardless of the message. In contrast, Zoomers perceive happiness depending on the underlying theme.

4.2.4 A Bunch Of Nothing

Still, on the topic of low-intensity emotions, this seems not to be completely constrained to the Zoomer's playlist. Some of the evidence points to a lack of emotional engagement with Millennials' songs, e.g., ''*Danceable... Has a lot less energy than the others (referring to the other songs on the Millennials playlist)*'(P2, "Quit Playing Games"), akin to some accounts regarding the Zoomer's playlist, e.g., '*Its danceable. With this type of song (referring to We Found Love), it is always the same thing*'(P4, "Titanium"). In tandem with low intensity, many accounts seem to point to an emotional dissociation with certain songs. This appears to lead to neutral responses in the listening accounts and the perceived emotion ones, with no clear distinction between generational groups.

Table 4. Results of coding process. The descriptions for each code are based on participant observations, with some codes being directly mentioned by one or more. Themes were found by aggregating codes with similar concepts.

Theme	Code	Description
Mirror of Society	Mirror of Society	An overarching topic found is the impact of the current state of society on music trends in both playlists. The Millennial's playlist focuses on the pursuit of romance and the suffering after a breakup, while in the Zoomer's playlist the message is about overcoming hardships and embracing your insecurities.
Disagreeing on Love	Suffer In the Name of Love	The underlying message of songs where this code was found is about the end of romantic relationships, and, more often than not, how hard it is for the artist to deal with it.
	High-Impact Vocals	Most of the songs that induced an emotion on the participants had some type of powerful vocals, many types accompanied by messages of sadness, pain or revolt.
	Paradox of Negative Emotions	Something that has been extensively studied in music psychology is how songs that convey powerful negative emotions evoke pleasurable feelings on listeners. Due to this paradox, most studies concerning musically-evoked emotions are constrained to perceived emotions.
Disagreeing on Low Intensity Emotions	Self-empowerement	The underlying message of these songs evokes feelings of self-worth in listeners and invites them to celebrate their insecurities.
	Digital Love	In more recent songs, extensive use of digital enhancement has become the norm in music production. Many accounts from both groups express the loss of essence derived from such heavy use, regarding the older songs as more soulful and more emotionally engaging.
	Instrumental-vocal Discordance	Some songs from the Zoomer's playlist produced accounts of confusion derived from the emotional discordance between the instrumental and vocal performance of certain songs.
A Bunch of Nothing	Emotional Disconnect	Many songs from both playlists failed to elicit any emotion when participants recounted their listening experiences, many attributing this to the formulaic construction and repeated ideas in them.
	Lack of Emotional Intensity	When describing their listening experience, participants from both generations mention the lack of actual emotion in the content of the song, which made them unsure of the emotion they perceived in the song.
Past Experiences Do Not Influence Judgement	Nostalgia Googles	This code encompasses all accounts of previous experiences that may influence the perception of emotion in a given song.
	Replaybility-based Aversion/Indifference	Many accounts, especially on music from the Zoomer's playlist, reveal a forced repeated listening of certain songs, which is reflected in the experience recounts as overly negative or neutral, disregarding the actual content of the song.
	Context-bound Experience	In some cases, the experience recount for a song was bound to some specific context, such as party music or music for dealing with an heartbreak, which may influence the perception of emotion.

4.2.5 Mirror Of Society

Taking into account the focus of this study, we expect to find some differences between the generations when asking for the perceived emotion of many songs, but the evolution of the Pop genre itself contributes even more to this disparity. We already discussed the different messages present in both sets of songs, and this theme points to the current state of society being reflected in these messages at those points in time, which we can see in statements such as '*In today's society, people are more individualistic... An awareness of ourselves (reflected in the Zoomers' playlist)*'(P3) and ''*The differences have much to do with social networks because you externalize your emotions more (in 2012 and beyond). This was more the role of music before (referring to the Milennials generation and before)*'(P5). A particular statement gives an interesting vision of the neutral responses from Millennials to most songs on the younger generation playlist, '*Songs of this era (referring to Zoomers) have more diversity... There is no pattern as demarcated as in the 90s... (There is) more diffusion and accessibility to music, which is good and not good, since more quantity does not mean better quality.*'(P2). Ultimately, all accounts related to this theme emphasize the differences between generations. In addition, the same accounts strongly suggest that changes in society, which trickle down to mainstream music, also have a substantial impact.

5. DISCUSSION

Before conducting this study, we expected to find differences in how different generations perceive emotions in music. These differences are likely due to each group's unique experiences during crucial moments in their lives and the prevailing trends and societal ideas at those times. However, we did not make assumptions about the specific factors contributing to these differences.

The unexpectedly low agreement rate may stem from the first author's perspective, being part of the youngest generation and aligning more closely with the predictions of those participants. Additionally, the author's expertise in

Music Emotion Recognition might have led to an oversight of the potential for many neutral responses when reporting the song's perceived emotion. This oversight is likely because research in this area often excludes songs without a clear dominant emotion.

The findings related to the factors contributing to such differences indicate that the generational group has some considerable impact. Considering Russell's model, we can note significant differences in how these groups perceive arousal when listening to songs that convey the theme of love-related suffering. Millennials tend to experience higher arousal, placing them in the quadrant associated with predominantly happier emotions. On the other hand, Zoomers tend to experience lower arousal, placing them in the quadrant associated with predominantly relaxed emotions.

Both groups generally agree on the valence of songs. This is also evident in "Set Fire To The Rain," a song from the Zoomer's playlist that conveys a similar message. As before, both groups agreed on the negative connotation but disagreed on the level of excitement. Millennials perceived higher excitement, primarily feeling angry emotions, while Zoomers felt lower excitement, primarily experiencing sad emotions.

The remaining songs from the Zoomers' playlists received mostly neutral responses from Millennials due to a perceived lack of intensity, as previously discussed. In contrast, Zoomers seemed to perceive a more pronounced emotion. Since this group has grown with songs presenting these traits and attended contexts that evoke the perceived emotions, like the party staples "We Found Love" and "Titanium", the accounts are more concrete, even when reporting that they are devoid of substance, akin to Millennials. However, these conclusions are not final due to the preliminary nature of the present study. Further investigation needs to be conducted with a larger sample size.

6. LIMITATIONS AND FUTURE WORK

The results we discussed are promising but have some limitations, especially concerning the number of participants. Both generational groups provided a great deal of evidence for the themes found, but some of the correlations were weak. Regarding the participants, there is a lack of diversity in this study sample. Only one participant identified as a woman in the Zoomers group, while all participants in the Millennial group identified as men.

Despite this, the latter has a better age distribution than the former, concentrating on the generation's beginning. Other individual characteristics seem to be well represented from the author's perspective. These issues should be addressed for more confidence in results from similar studies. Regarding data collection, the interviews could have provided more information had they been taped. Since the study follows a phenomenological approach, physical reactions to the songs could have also provided more evidence for the themes found.

The most glaring limitation is the set of playlists used for conducting this study. Since they are based on one particular moment in time, they hardly represent the entirety of two generations. Addressing this point would be another study altogether, as already mentioned in Section 3.2. Still, in future work, it would be interesting to extend this study to other generations and understand if similar themes emerge since some of the found themes seem to be independent of generation, namely the ones discussed in Sections 4.2.1, 4.2.4 and 4.2.5.

Another interesting approach would be to conduct this study with other genres. However, this may be challenging since some genres are not typically appealing to the masses or are less emotional than Pop generally is. From the output of this work, there is some interest in understanding the factors behind the more concrete accounts of perceived emotion from Zoomers regarding their generational playlist compared with Millennials and to what extent the differences in arousal perception can be observed.

7. CONCLUSIONS

Music psychology has long studied differences in the perception of emotion in music. However, the focus on differences between generational groups and corresponding music trends is limited.

Six participants from two generational groups, i.e., Millennials and Zoomers, listened to two playlists representing their generation's music trends to explore the differences and contributing factors. After recounting their listening experience, participants were asked to identify the predominant emotion they perceived in each song. The reported emotional perceptions were mostly uniform within each generation but more varied between groups than expected initially.

Various common themes emerged after conducting an open coding approach to the transcribed interviews. One notable finding was the discrepancy in the energy conveyed by songs with a message about suffering for love. Zoomers tended to perceive more pronounced emotions in songs from their playlist, while some reported a lack of emotion during the listening experience across both generations.

These findings point to evolving perceptions of emotion in music in successive generations and considerable differences in music trends. There are differences not only regarding the participant generation but also in the songs themselves. Future work should expand the sample size of each group to consolidate these findings.

Acknowledgments

This work is funded by FCT - Foundation for Science and Technology, I.P., within the scope of the projects: MERGE - PTDC/CCI-COM/3171/2021 financed with national funds (PIDDAC) via the Portuguese State Budget; and project CISUC - UID/CEC/00326/2020 with funds from the European Social Fund, through the Regional Operational Program Centro 2020. Renato Panda was supported by Ci2 - FCT UIDP/05567/2020.

8. REFERENCES

[1] Y.-H. Yang, Y.-F. Su, Y.-C. Lin, and H. H. Chen, "Music emotion recognition: The role of individuality," in *Proceedings of the International Workshop on Human-Centered Multimedia*, 2007, pp. 13–22.

[2] X. Hu and J. H. Lee, "A cross-cultural study of music mood perception between american and chinese listeners," in *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012, pp. 535–540.

[3] T. Green, "muzic appreciation: Bridging the gap between generation z & classical music," 2018. [Online]. Available: https://digitalcommons.liberty.edu/masters/490/

[4] K. Pyrovolakis, P. Tzouveli, and G. Stamou, "Multimodal song mood detection with deep learning," *Sensors*, vol. 22, pp. 387–392, 2009.

[5] P. N. Juslin and P. Laukka, "Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening," *Journal of New Music Research*, vol. 33, no. 3, pp. 217–238, 2004.

[6] A. Gabrielsson, "Emotion perceived and emotion felt: Same or different?" *Musicae Scientiae*, vol. 5, no. 1, pp. 123–147, 2001.

[7] A. Pannese, M.-A. Rappaz, and D. Grandjean, "Metaphor and music emotion: Ancient views and future directions," *Consciousness and Cognition*, vol. 44, pp. 61–71, 2016.

[8] K. Hevner, "Experimental studies of the elements of expression in music," *The American Journal of Psychology*, vol. 48, pp. 246–268, 1936.

[9] J. Posner, J. A. Russell, and B. S. Peterson, "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology," *Development and Psychopathology*, vol. 17, pp. 715–734, 2005.

[10] T. Eerola and J. K. Vuoskoski, "A comparison of the discrete and dimensional models of emotion in music," *Psychology of Music*, vol. 39, no. 1, pp. 18–49, 2011.

[11] A. Aljanaki, Y.-H. Yang, and M. Soleymani, "Developing a benchmark for emotional analysis of music," *PLOS ONE*, vol. 12, no. 3, pp. 1–22, 2017.

[12] C. P. Heinlein, "The affective characters of the major and minor modes in music." *Journal of Comparative Psychology*, vol. 8, no. 2, p. 101, 1928.

[13] K. Hevner, "The affective character of the major and minor modes in music," vol. 47, no. 1, pp. 103–118, 1935.

[14] T. Fritz, S. Jentschke, N. Gosselin, D. Sammler, I. Peretz, R. Turner, A. D. Friederici, and S. Koelsch, "Universal recognition of three basic emotions in music," *Current Biology*, vol. 19, no. 7, pp. 573–576, 2009.

[15] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*, vol. 6, no. 3, pp. 169–200, 1992.

[16] J. W. Creswell and C. N. Poth, *Qualitative inquiry and research design: Choosing among five approaches*. Sage Publications, 2016.

EXPLORING THE EMERGENCE OF BEAT INDUCTION USING A 'SWARM OF ONSETS' GENERATIVE MODEL

Nolan LEM [1] and **Jens AHRENS**[1]

[1]*Division of Applied Acoustics*, **Chalmers Institute of Technology**, Gothenburg, Sweden

ABSTRACT

This study investigates the emergence of the beat percept in auditory textures that are built up from homogeneous sources. Sensorimotor synchronization (SMS) models of beat perception have examined how humans synchronize to musically relevant signals, however there have been few systematic inquiries into SMS in more soundscape-oriented phenomena where a listener is confronted by multiples of concurrent, interactive sources. In this experiment, participants were asked to synchronize (via tapping) to stimuli created from a generative model where forty metronome sounds are clustered around periodic, temporal centers using a Gaussian probability distribution function parameterized with eight levels of variance to produce 'swarms' of temporally aligned onsets. Participants encountered stimuli both in mono (dichotic) and binaural spatial presentations as we hypothesized that the spatial presentation might produce significant differences in SMS performance. As evidenced by inter-beat interval analysis and phase coherence analysis, participants were able to largely synchronize to six of the eight onset variance conditions despite tap frequency increasing with onset spread. Similarly, while we found a slight interaction between spatial format and stimuli onset variance, no significant differences were found between the overall tap responses to the binaural versus mono stimuli. This study is a first look into how the beat percept arises to sound textures whose constituent elements contain different levels of embedded synchrony.

1. INTRODUCTION

To be immersed in a sounding environment is to encounter and listen to many different collections of sounds that make up the auditory scene. By segregating and identifying the multitude of sources contained within the soundscape, we make sense of the auditory environment around us. For example, we can detect and isolate sound objects, appreciate complex musical phenomena, and identify prospective threats, all while forming a coherent auditory map of our surroundings that allow us to navigate a physical space [1–3].

While much research in sensorimotor synchronization (SMS) has investigated the human ability to entrain, synchronize, and coordinate behavior with a variety of musically-oriented signals, there has not been a systematic inquiry into how SMS emerges from more ambient, naturalistic sounds. Naturalistic auditory textures, such as rain, hail, crackling fire, stridulating insects, or the rustle of wind, are often made up of sounding elements that interplay with one other in ways both complex and predictable [4]. Here, variegated interactions at the local level can give rise to collective behaviors that result from different cyclic patterns in nature where biological materials respond to environmental demands. In short, different behavioral dynamics create sound in adaptive ways and this has been reflected in a number of studies such as insect communication, crowd dynamics, crowd applause, and traffic noise [5–8].

Taken as a class of auditory signals, many naturalistic sound textures exhibit signal stationarity which is associated with specialized forms of auditory processing [9]; McDermott et al. successfully recreated such sound textures by generating signals that mimic their statistical properties via cochlea filtering during natural listening experiences. Since sound textures are comprised of large numbers of concurrent sources, this line of research suggests perceptual systems may rely on forms of temporal averaging in order to recognize and detect such sounds. Other generative models for soundscape recreation take a different approach by synthesizing individual sound sources en masse, exploring techniques such as granular and concatenative synthesis, physical modeling, and other procedural algorithms that are more physically analogous to sound production in the real world (For a review, please see [4]). If individual sound sources interact and synchronize their actions, how does collective behavior shape the resulting soundscape? Human synchronization can offer insights to better understand how we might perceive and engage with these auditory worlds.

We can listen for auditory cues in the form of repeated sonic events in order to sense regularity and construct a notion of pulse, a sensorimotor skill that is critical for the performance of music and group synchronization in general [10, 11]. The well-known 'cocktail party effect' provides evidence that we are able to 'listen in' and latch onto constituent elements in a dense, auditory textures comprised of multiple homogeneous sources [12, 13]. Even within an auditory stream of spoken language, we have to parse linguistic signals through various mechanisms such as synchronizing to phoneme structure [14, 15]. In one study, short repetitions of synthetic soundscapes were re-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

peated at different intervals to determine if listeners were able to detect this embedded regularity [16]. The authors found that recurring segments of naturalistic sounds facilitated the detection of sound examples within multiple mixtures and that listeners may unconsciously learn priors about a sound texture via repetition. Furthermore, there is limited research on how source spatialization affects the beat percept and SMS in general. While large-scale, immersive speaker arrays may provide more dynamic and enveloping listener experiences at the expense of degrading the perception of rhythmic information [17], it is not clear what effect listener envelopment has on the emergence of the beat percept from otherwise stationary signals.

This perceptual study examines how the beat percept arises in sequences of concurrent sound onsets that are synchronized at different levels of temporal alignment. How synchronized do sound events have to be in order for humans to notice a sense of regularity, latch onto an entraining rhythm, and coordinate behavior in tandem with the incipient beat? By creating auditory textures built up from superpositions of sound onsets at different levels of synchrony, we can examine the point at which we begin to induce beat via tap synchronization. As such, this study is a first look into how the beat percept might arise within auditory textures comprised of homogeneous sound sources that are temporally allocated as (Gaussian) random variables over periodic moments in time. We hypothesized that people would use the embedded onset densities or 'swarms' as an auditory cue in order to synchronize to such auditory textures. Similarly, we anticipated that the spatial format of such sounds would influence human synchronization performance.

2. METHODS

2.1 Stimuli Preparation

Audio stimuli were generated from a generative Gaussian probability density function (PDF) that distributes 40 independent onsets in the form of metronome clicks around 15 "beat centers" as derived from 5 tempo conditions (60,68,77,88,100 bpm). For each tempo, the PDF was centered at the time associated with the period of each tempo condition, parameterized with 8 levels, or "gradations", of variance that correspond to a set of standard deviations spanning a linear range from 0.0 to 0.6 of the temporal period. Depending on the variance, this generative PDF allocates a 'swarm of onsets' more or less tightly around periodic centers corresponding to the tempo of each condition (each of the gradation conditions of variance scaled with tempo so as to keep the spread of distributions proportionate across the tempo condition). An audio sample of a single metronome click was used as the auditory source onset. Stimulus generation is illustrated in the Figure 1 which shows the Gaussian PDF parameterized at the 8 levels of variance for a single tempo condition (the lowest variance condition represents the isochronous case). Kernel Density Estimates (shown in orange) of the onset histograms are shown over each of the 15 beat centers for each variance condition. Correspondingly, the resultant audio waveform

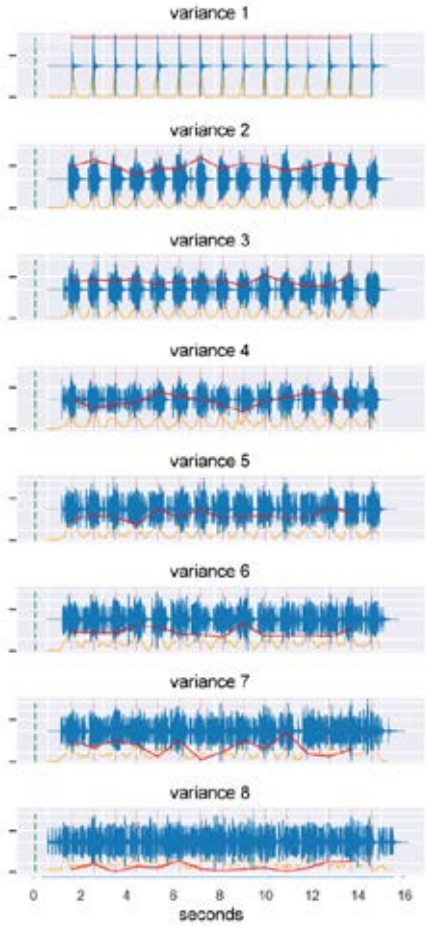


Figure 1. Examples of the stimulus waveforms for each of the variance conditions, or "gradations", for a single tempo condition. Kernel density estimates (shown in orange) approximate the shape of the generative Gaussian PDF centered on each of the 15 beat centers (shown as vertical red lines). The stimulus phase coherence, R_{onsets}, sampled at each temporal beat center, are shown in red.

is overlayed onto this plot; the red line shows the phase coherence magnitude (see Equation 1) , R_{onsets}, of the stimuli which is a summary statistic which indicates the global synchrony of the onsets over each beat region.

When the stimuli variance is low, the metronomes sounds are tightly distributed around the period associated with the tempo; conversely, when variance is high, the samples are more randomly distributed throughout the beat-to-beat temporal region. During stimulus creation, two versions of each variance gradation and tempo condition were created resulting in 80 stimulus sequences.

2.2 Test Procedure

14 volunteers from the University community participated in this study. Participants completed a questionnaire before the experiment began which collected demographic data. 4 women participated in the study and the mean age was 30.5 yrs (SD(age) = 11.2 yrs). 2 of the participants were left-handed and the entire cohort had 4.8 yrs as the average years of musical training (SD(musical training) = 6.2 yrs).

Participants were asked to tap on the spacebar of a 15" MacBook Pro Laptop using the dominant finger of their dominant hand with Sony MDR7506/1 headphones. The experimental stimuli were created using the SuperCollider music programming language. The stimuli were presented in randomly in two blocks, each block either consisting of sound sequences that were either dichotic (i.e., mono, the same sound presented to both ears) or binaurally using the SoundScape Renderer v. 0.6.1 with a head tracking unit (Supperware Head Tracker 1) [18]. The 40 onsets were distributed onto 40 virtual source positions located on the horizontal plane as a circle around the listener.

First, we collected a 15 second period of participant spontaneous tempo by asking them to tap at a rate that was comfortable to them without any auditory stimuli. Following this, participants attempted a practice block consisting of 4 auditory stimulus sequences before each experimental block. They proceeded onto the experimental tapping task which consisted of 80 audio files in each block which resulted in a total duration of around 20 minutes per block. The total experiment took approximately 40 minutes to complete.

2.3 Data Analysis

The parameters associated with each stimulus—namely the variance and the tempo — were initially collected and saved to disk during stimulus generation. The participant tap responses were analyzed in terms of their inter-tap-interval and the phase coherence magnitude, (R), and angle, (ψ). The inter-tap interval (ITI) is defined as the amount of time in between each of their successive taps which can be normalized with respect to the stimuli period (obtained from the tempo condition) so as to compare tap responses between tempo conditions; this is referred to as the normalized ITI (nITI). A nITI of 1 would mean that the participant tapped at the same rate as the stimulus beat centers (conversely, a nITI < 1 would indicate taps that occurred at a rate that was faster than the stimulus beat centers).

The phase coherence analysis is determined from calculating the stimulus onsets and participant tap response complex order parameters from which the phase coherence magnitude, R, and average angle, ψ, are obtained from Equation 1 where N is the number of onsets and ϕ_i is the phase in radians between consecutive beat centers (j is a complex number). R is a value between 0 and 1 which represents the synchrony of the group of onsets and ψ (0-2π) represents the average angle among all onsets.

$$R\, e^{j\psi} = \frac{1}{N}\sum_{i=1}^{N} e^{j\phi_i} \tag{1}$$

Unlike ITI analysis, phase coherence analysis provides information on how taps line up with the center of the beat and provides information about how taps may tend to precede ('lead') or fall behind ('lag') the center of the beat. Each participants tap and stimulus onset can be mapped to a point on a circle where the bounds of the circle represent the distance between consecutive beat centers where center of the beat is defined to be at π (onsets or taps $> \pi$ imply a lag and tap responses $< \pi$ imply a lead). For the tap and stimuli onset distribution, we can create a phasor with magnitude, R, at average angle, ψ, that reflects both the aggregate alignment of the participants' tap responses and the stimuli onsets. Rayleigh tests were performed to determine if the tap and onset distributions contained any significant directionality and lend support to the idea that the phase coherence magnitude and angle can be considered to be valid [19]. Similarly, comparisons between distributions can be made using Watson-Wheeler Tests, a circular statistic that tests for homogeneity on two or more samples of circular data in terms of the mean or variance [20].

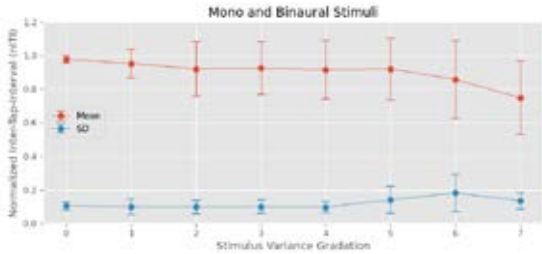


Figure 2. Participant mean (red) and SD (blue) of the normalized inter-tap-interval (nITI) over each stimulus variance condition. Error bars indicate the mean subject error.

3. RESULTS

Figure 2 shows the participant mean nITIs and SD of nITIs for each of the stimuli variance gradations (1-8) for the combined mono and binaural sounds. Here we see how nearly all of the participants were able to tap with a nITI≈1 up until the 7th and 8th variance gradation at which point there is decrease in nITI which implies that the participants began to tap at a faster rate than the stimulus beat centers.

Figure 3 shows the mean nITI and SD distribution of all tapping trials as a function of the stimuli phase coherence. As the phase coherence of the stimuli decreases (at $R \approx 0.3$), many participants began to tap at a faster rate (lower nITI) with much more tap-to-tap variance (larger SD of nITI) across participants.

Table 1 summarizes the results of the three-way repeated measures ANOVA of the stimulus variance, spatial presentation, and tempo conditions for the mean nITI and SD respectively. For the mean nITI, the ANOVA revealed significant main effects of stimuli variance, tempo, and an interaction between stimuli variance and tempo. The main effect of stimuli variance was significant (F(7,84) = 5.84, $p < 0.001$) where post-hoc tests showed mean nITI progressively decreased from the stimulus variance gradation 1 to 8 (0.98, 0.95, 0.92, 0.93, 0.92, 0.92, 0.86, 0.75) and post-hoc analysis showed more significant mean nITI values as compared to higher stimulus variance gradations.

For the SD of the nITIs, the ANOVA revealed significant main effects only for the stimulus variance (F(7,84) = 3.83, $p = 0.001$) from gradation 1 to 8 (0.11, 0.10, 0.10, 0.10, 0.10, 0.14, 0.18, 0.14) where the mean SD of the subject

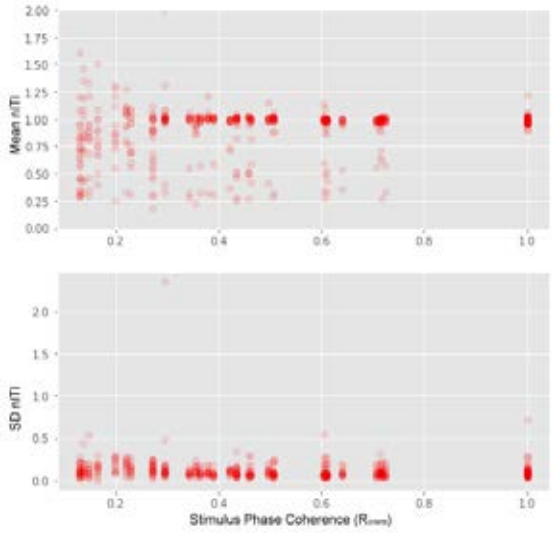


Figure 3. All mean (top) and SD (bottom) of the participant nITIs for each trial plotted as a function of the stimulus phase coherence. Mean nITIs < 1.0 imply that a participant tapped at a rate that was faster than the rate associated with the stimulus beat centers.

	Mean nITI	SD of nITI
Factor		
Variance Gradation	1-(3*,4*,5**,7***,8***) 2(-7*,8***) (3,4,5,6,7)-8***	2-8*** 2-(7*,8**) 3-(7*,8**) 4-(7*,8**) 5-(7*,8***)
Spatial Presentation	NS	NS
Tempo	60-(68*,77***,88***,100***) 68-100**	NS
Tempo x Gradation	Var 6: 60-(88*,100*) Var 7: 60-(77*,100**) Var 8: 60-(77*, 88*, 100*) 68-100**	NS

Table 1. Results of the three-way ANOVAs for the mean and SD of the participant nITIs. Note. Significance levels are indicated as *** $p < .001$, ** $p < .01$, * $p < .05$, NS = not significant. The levels for the variance conditions are (1-8) and the five levels for the tempo conditions were (60,68,70,88,100).

taps tended to increase over stimulus gradation.

Lastly, the main effects of the tempo on the nITI was significant (F(1,12) = 27.4, $p < 0.001$) and the results of the post-hoc tests indicated that tap nITI tended to increase with increasing tempo conditions; post-hoc results showed bpm 60, 68, 77, 88, and 100 corresponding to 0.83, 0.89, 0.92, 0.93, 0.96 nITI respectively.

The phase coherence of the taps and the stimulus onsets were computed to derive a phasor representing the average angle across participants with respect to the center of the beat. These phase coherence distribution plots are shown for each variance gradation in Figure 4. Rayleigh tests reported all of the tap and stimulus onset distributions were shown to have significant directionality with the exception of the tap distribution of variance gradation 8. Watson-Wheeler tests showed significant mean differences between the distributions of the tap and onset distributions for every gradation (all comparisons per variance gradation

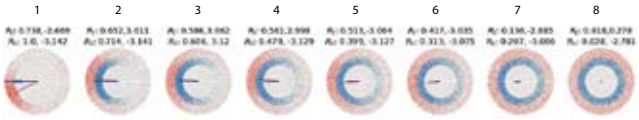


Figure 4. Phase coherence "swarm" plots showing the distribution of the participant taps (red) versus the stimulus onsets (blue). Phase coherence magnitude and average angle (R, ψ (rad)) are shown as phasors for the taps (R_t) and onsets (R_o).

$p < 0.0001$, except variance gradation 8, ($p = 0.003$)).

4. DISCUSSION

We observe three central tendencies among participant tap responses: 1.) all participants tapped in line with the region of the stimuli containing the highest onset density (the beat center) across variance gradation and 2.) participants slightly increased the mean frequency of their taps when synchronizing to stimuli with increased onset variance but tapping was slower at higher tempos and 3.) the spatial presentation (mono, or binaurally rendered) of the onset swarms did not significantly affect their tap responses. These results indicate that synchronization performance only degraded at gradation 7 and 8 suggesting that up until that variance gradation, participants were able to induce a beat percept from the stimulus sequences. While SMS probably depends on the source sound material and numerosity of sources and events, this study suggests that sound onsets normally distributed in time can induce the beat percept if they are sampled from a normal distribution with a SD that is ≈ 0.3 of the temporal period.

Previous research has observed tapping rate increases with stimulus asynchrony which might be a result of individual differences [21]. Surprisingly, participants taps were generally consistent throughout the course of the stimuli sequences since tap-to-tap variance (SD of nITI) was generally low and only significantly increased toward the last two stimuli variance gradation. While it is possible that subjects simply tapped at consistent rate without being entrained, their tapping rates during stimulus presentation were substantially faster than their collected spontaneous rates. For the last two variance gradations, approximately 40% of their taps still fell within a 10% window and 20% window of the target stimulus tempo for the last two stimulus variance gradations respectively. Why might people tap at faster rate when they lose the beat percept? Once individual onsets are no longer grouped as belonging to a particular beat center, individual onsets—taken sequentially rather than streamed — may begin to dominate the perception of the beat [22, 23]. Similarly, faster tapping might reflect strategies that incorporate active sensing where humans might sample from sensory spaces at a faster rate [24, 25].

No significant differences were found in tap responses to the mono (dichotic) presented sounds versus the binaurally presented ones. Initially, we hypothesized that the listener being enveloped by the sounds might allow them to segre-

gate specific sounds in the mix and use that sound as an auditory cue for synchronization. However, our findings support previous research showing how only four spatialized speakers can approximate spatial envelopment for broadband, stationary sounds such as those found in naturalistic soundscapes [26, 27].The findings of this study support the notion that SMS performance is not facilitated nor constrained by the presence of more spatialized sources.

The phase coherence angle of the group's tapping response to gradation 1, a completely isochronous stimuli, shows a lag (≈0.47 rad, ≈75 ms for the 60 bpm condition, ≈45 ms for the 100 bpm condition) between the mean of the subject tap and the center of the beat is well within the range of negative mean asynchrony (NMA) observed in other SMS studies which for humans to coordinate their taps with a slight delay behind an entraining stimulus onset [28]. ψ_{taps} slightly preceded ψ_{onsets} during variance gradation 2, 3, and 4 suggesting that listeners may have been more adaptive in their tapping; error correction models have accounted for such anticipatory responses in auditory pacing sequences with expressively modulating tempos [29]. However, as the stimulus variance increases, we observed a reduction of phase coherence magnitude across and within participant tap responses. Surprisingly, while the R_{taps} value remained small (≈ 0.14), ψ_{taps} still remains pointing towards the center of the beat until the last variance gradation which suggests that overall, there was still a tendency to tap toward either the center of the beat or in between two beat centers. Another indication of the participant entrainment is observed in the transition from variance 4 to 6, where $R_{taps} > R_{onsets}$: the participants placed their taps with more temporal alignment than the onsets themselves.

Ultimately, our findings support the notion that onset density is indeed used by the listener in order to construct a sense of beat, however tentative that may be.

It should be noted that the waveform of the stimuli associated with larger variances did contain a weakly periodic amplitude envelope suggesting that volume might be the entraining auditory cue as shown in other research in amplitude modulated noise perception [30] but that the majority of participants were still able to coordinate their taps with onset density when phase coherence magnitude, R, was small (≈0.3). Even among a backdrop of a homogeneous auditory texture, we suspect that participants are able to track the accumulation of onsets as they coalesce around temporal centers and that the auditory mechanism involved in auditory averaging of stationary signals transitions to parsing onsets into periodic temporal regions, a process that shares many features with beat-bin based models of microtiming in groove-based music [31]. In fact, this model is built into to the stimuli generation a priori: PDFs generate onsets that align with more or less spread around temporal centers, and beat perception in this context may rely on integrating onsets within durational boundaries that arise from making causal inferences from priors in order to structure beat relevant information in the real-time [32].

The SMS behaviors observed in this study might not apply to phenomena exhibiting other statistical distributions given that our stimuli produced onset swarms normally distributed in time; natural phenomena exhibiting other statistical distributions might induce other forms of synchronization. This analysis did not capture any aspects related to the time structure of participant taps—that is, there was no analysis to determine if their taps evolved over the course of the stimuli sequences. We observed that tap variance was relatively low for most of the variance gradations but further inquiry into the nature of when participants began to tap or how their taps evolved on a tap-to-tap basis would be interesting to evaluate more thoroughly. While we did capture the participants' head tracking movement data during the experiment, we noticed that they did not tend to move their head during the tapping task and therefore did not analysis this data. Because the 40 virtual sources sufficiently enveloped the listener, head movements would not likely have facilitated beat induction.

5. CONCLUSIONS

This study provided a first look into how a sense of beat might arise from interactions among sound sources that makeup a larger auditory texture. We used a probability density function parameterized at different levels of variance to span a range of synchronous onset sequences. While spatial source location did not significantly affect tap performance, ITI and phase coherence analysis suggested that participants synchronized their behavior with the recurring onset density and that the gradual loss of the beat percept was associated with tapping at an accelerated rate. Further inquiry into SMS using onset distributions might examine the effects of source numerosity; for example, listeners may well use such spatialized cues when the number of perceived sound sources in a space is lower. This would be particularly true if the sources themselves were less homogeneous in timbre, as more salient sound objects might direct attention in such a way as to significantly alter their SMS performance [33, 34].

Acknowledgments

This research was supported by the Anna Ahrenberg Foundation.

6. REFERENCES

[1] A. J. King, "Sensory experience and the formation of a computational map of auditory space in the brain," *BioEssays*, vol. 21, no. 11, pp. 900–911, Oct. 1999. [Online]. Available: https://onlinelibrary.wiley.com/doi/10.1002/(SICI)1521-1878(199911)21:11⟨900::AID-BIES2⟩3.0.CO;2-6

[2] K. T. Gagnon, M. N. Geuss, and J. K. Stefanucci, "Fear influences perceived reaching to targets in audition, but not vision," *Evolution and Human Behavior*, vol. 34, no. 1, pp. 49–54, Jan. 2013. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S1090513812000918

[3] C. Douglas, J. Noble, and S. McAdams, "Auditory Scene Analysis and the Perception of Sound Mass in Ligeti's Continuum," *Music Perception: An Interdisciplinary Journal*, vol. 33, no. 3, pp. 287–305, 2016.

[4] D. Schwarz, "State of the art in sound texture synthesis," *Proceedings of the International Conference on Digital Audio Effects, DAFx*, no. May, pp. 221–232, 2011, iSBN: 9782954035109.

[5] M. Greenfield, "Cooperation and conflict in the evolution of signal interactions," *Annual Review of Ecological Systems*, vol. 25, pp. 97–126, 1994.

[6] M. Moussaïd, D. Helbing, and G. Theraulaz, "How simple rules determine pedestrian behavior and crowd disasters," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 17, pp. 6884–6888, 2011.

[7] Q. Meng and J. Kang, "The influence of crowd density on the sound environment of commercial pedestrian streets," *Science of the Total Environment*, vol. 511, pp. 249–258, 2015, iSBN: 1759-4685 (Electronic)\r1759-4685 (Linking).

[8] Z. Neda, E. Ravasz, T. Vicsek, Y. Brechet, and A. Barabasi, "Physics of the rhythmic applause," *Theoretical and Mathematical Physics*, vol. 104, no. 3, pp. 1104–1107, 1995, iSBN: 1063-651X.

[9] J. H. McDermott, M. Schemitsch, and E. P. Simoncelli, "Summary statistics in auditory perception," *Nature Neuroscience*, vol. 16, no. 4, pp. 493–498, 2013, iSBN: 1546-1726 (Electronic)\r1097-6256 (Linking).

[10] E. W. Large and C. Palmer, "Perceiving temporal regularity in music," *Cognitive Science*, vol. 26, pp. 1–37, 2002.

[11] B. H. Merker, G. S. Madison, and P. Eckerdal, "On the role and origin of isochrony in human rhythmic entrainment," *Cortex*, vol. 45, no. 1, pp. 4–17, 2009, publisher: Elsevier Srl ISBN: 0010-9452. [Online]. Available: http://dx.doi.org/10.1016/j.cortex.2008.06.011

[12] S. R. Alain, Claude; Arnott, "Selectively attending to auditory objects," *Frontiers in Bioscience*, vol. 5, no. 3, p. A505, 2000, iSBN: 1093-4715 (Electronic) 1093-4715 (Linking). [Online]. Available: https://www.bioscience.org/2000/v5/d/alain/list.htm

[13] J. Xiang, J. Simon, and M. Elhilali, "Competing streams at the cocktail party: Exploring the mechanisms of attention and temporal integration," *Journal of Neuroscience*, vol. 30, no. 36, pp. 12 084–12 093, 2010, iSBN: 9550091023.

[14] J. Morton, S. Marcus, and C. Frankish, "Perceptual centers (P-centers)," *Psychological Review*, vol. 83, no. 5, pp. 405–408, 1976.

[15] R. C. Villing, B. H. Repp, T. E. Ward, and J. M. Timoney, "Measuring perceptual centers using the phase correction response," *Attention, Perception, and Psychophysics*, vol. 73, no. 5, pp. 1614–1629, 2011, iSBN: 1341401101101.

[16] J. H. McDermott, D. Wrobleski, and A. J. Oxenham, "Recovering sound sources from embedded repetition," *Proceedings of the National Academy of Sciences*, vol. 108, no. 3, pp. 1188–1193, 2011, iSBN: 0027-8424. [Online]. Available: http://www.pnas.org/cgi/doi/10.1073/pnas.1004765108

[17] T. Mouterde, N. Epain, S. Moulin, and E. Corteel, "On the factors influencing groove fidelity in immersive live music events," *AES Convention*, 2023.

[18] J. Ahrens, M. Geier, and S. Spors, "The soundscape renderer: A unified spatial audio reproduction framework for arbitrary rendering methods," in *Audio Engineering Society Convention 124*, May 2008. [Online]. Available: https://www.aes.org/e-lib/browse.cfm?elib=14460

[19] B. R. Moore, "A Modification of the Rayleigh Test for Vector Data," *Biometrika*, vol. 67, no. 1, p. 175, Apr. 1980. [Online]. Available: https://www.jstor.org/stable/2335330?origin=crossref

[20] K. V. Mardia, "Statistics of Directional Data," *Journal of the Royal Statistical Society: Series B (Methodological)*, 1975.

[21] N. Lem and T. Fujioka, "Individual differences of limitation to extract beat from Kuramoto coupled oscillators: Transition from beat-based tapping to frequent tapping with weaker coupling," *PLOS ONE*, vol. 18, no. 10, p. e0292059, Oct. 2023. [Online]. Available: https://dx.plos.org/10.1371/journal.pone.0292059

[22] A. Bendixen, S. L. Denham, K. Gyimesi, and I. Winkler, "Regular patterns stabilize auditory streams," *The Journal of the Acoustical Society of America*, vol. 128, no. 6, pp. 3658–3666, Dec. 2010. [Online]. Available: https://pubs.aip.org/jasa/article/128/6/3658/904282/Regular-patterns-stabilize-auditory-streams

[23] R. Brochard, C. Drake, M.-C. Botte, and S. McAdams, "Perceptual organization of complex auditory sequences: Effect of number of simultaneous subsequences and frequency separation." *Journal of Experimental Psychology: Human Perception and Performance*, vol. 25, no. 6, pp. 1742–1759, Dec. 1999. [Online]. Available: https://doi.apa.org/doi/10.1037/0096-1523.25.6.1742

[24] S. C.-H. Yang, D. M. Wolpert, and M. Lengyel, "Theoretical perspectives on active sensing," *Current Opinion in Behavioral Sciences*, vol. 11, pp. 100–108, Oct. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352154616301255

[25] C. E. Schroeder, D. A. Wilson, T. Radman, H. Scharfman, and P. Lakatos, "Dynamics of Active Sensing and perceptual selection," *Current Opinion in Neurobiology*, vol. 20, no. 2, pp. 172–176, Apr. 2010. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0959438810000322

[26] O. Santala and V. Pulkki, "Directional perception of distributed sound sources," *The Journal of the Acoustical Society of America*, vol. 129, no. 3, pp. 1522–1530, 2011. [Online]. Available: http://asa.scitation.org/doi/10.1121/1.3533727

[27] T. Pihlajamäki, O. Santala, and V. Pulkki, "Synthesis of Spatially Extended Virtual Sources with Time-Frequency Decomposition of Mono Signals," *J. Audio Eng. Soc.*, vol. 62, no. 7, 2014.

[28] G. Aschersleben and W. Prinz, "Synchronizing actions with events: The role of sensory information," *Perception & Psychophysics*, vol. 57, no. 3, pp. 305–317, Apr. 1995. [Online]. Available: http://link.springer.com/10.3758/BF03213056

[29] M. C. Van Der Steen, N. Jacoby, M. T. Fairhurst, and P. E. Keller, "Sensorimotor synchronization with tempo-changing auditory sequences: Modeling temporal adaptation and anticipation," *Brain Research*, vol. 1626, 2015.

[30] N. F. Viemeister, "Temporal modulation transfer functions based upon modulation thresholds," *The Journal of the Acoustical Society of America*, vol. 66, no. 5, pp. 1364–1380, Nov. 1979. [Online]. Available: https://pubs.aip.org/jasa/article/66/5/1364/778563/Temporal-modulation-transfer-functions-based-upon

[31] A. Danielsen, K. Nymoen, E. Anderson, G. S. Câmara, M. T. Langerød, M. R. Thompson, and J. London, "Where is the beat in that note? Effects of attack, duration, and frequency on the perceived timing of musical and quasi-musical sounds," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 45, no. 3, pp. 402–418, 2019.

[32] M. T. Elliott, A. M. Wing, and A. E. Welchman, "Moving in time: Bayesian causal inference explains movement coordination to auditory beats," *Proceedings of the Royal Society B: Biological Sciences*, vol. 281, no. 1786, 2014.

[33] M. Elhilali, J. Xiang, S. A. Shamma, and J. Z. Simon, "Interaction between attention and bottom-up saliency mediates the representation of foreground and background in an auditory scene," *PLoS Biology*, vol. 7, no. 6, 2009.

[34] S. A. Shamma, M. Elhilali, and C. Micheyl, "Temporal coherence and attention in auditory scene analysis," *Trends in Neurosciences*, vol. 34, no. 3, pp. 114–123, 2011, publisher: Elsevier Ltd ISBN: 1878-108X (Electronic) 0166-2236 (Linking). [Online]. Available: http://dx.doi.org/10.1016/j.tins.2010.11.002

PAPER & POSTER /DEMO SESSION 5

/ Beamix: Application of an Ambisonics Filter Bank for the Creation of Directional Effects

/ Toward a Novel Set of Pinna Anthropometric Features for Individualizing Head-Related Transfer Functions

/ A Speaker Agnostic Approach to Spatialisation in Electroacoustic Music

/ Csound vs. ChucK: Sound Generation for XR Multi-Agent Audio Systems in the Meta Quest 3 Using the Unity Game Engine

/ Comparing Four 360-Degree Cameras for Spatial Video Recording and Analysis

/ Exploring Convolutional Neural Networks for Multimodal Classification of Expressive Piano Performance

/ Combining Hand Pose Estimation with Audio for Noise-robust Piano Score Following

/ IEMI: an Immersive, Evolving Motion-controlled Installation

/ A Cover Image Composition Method for Music Content Using Multimodal Image Retrieval and Cropping Techniques

BEAMIX : APPLICATION OF AN AMBISONICS FILTER BANK FOR THE CREATION OF DIRECTIONAL EFFECTS

Thibaut CARPENTIER (thibaut.carpentier@ircam.fr) [1]

[1] **STMS Lab, IRCAM — CNRS — Sorbonne Université — Ministère de la Culture**, 1, place Igor-Stravinsky, Paris, 75004 France

ABSTRACT

In this article we present `beamix`, an audio engine and its graphical user interface which allow one to produce directional audio effects (FX), that is, sound transformations localized in certain areas of space. These effects (such as equalizer, compressor, etc.) are applied to an Ambisonics-encoded sound scene, which was previously decomposed by a spatial filter bank. The latter, based on the state of the art, applies beamforming with virtual microphones (a process analogous to a plane-wave decomposition of the sound field) so as to partition the 3D or 2D sound field into a set of angular sectors. A chain of audio effects can be inserted into each of these sectors, allowing fine manipulation of the soundscape. The approach thus proposed offers more flexibility than transformations applied in the spherical harmonics domain, in particular for the realization of non-linear effects. The tool is typically useful for production or post-production of Ambisonics scenes, but can also be used for editing directional room impulse responses (DRIR).

1. INTRODUCTION

1.1 Filter bank

In signal processing, a filter bank refers to a set of K band-pass filters that decompose an input signal x into K components, each representing a frequency sub-band. This is ubiquitously used in the field of computer music to achieve *multiband* effects [1, Section 2.6.3], such as graphic equalizers (EQ) or multiband dynamic range compression (compressor/expander). An important property of filter banks is their ability correctly to *reconstruct* the input signal. Perfect reconstruction is achieved when the output signal y is such that $y[n] = x[n]$ when no effect is applied.

1.2 Spatial filter bank

A *spatial* filter bank is conceptually similar, except that the input signal x is not a monophonic signal but a spatial sound field, and the filtering operates in the angular domain (azimuth and/or elevation angles) rather than in the frequency domain. In other words, a spatialized sound scene — represented in 3D on the surface of a sphere, or in 2D on a circle — is decomposed into a set of sectors forming a partition of space. Filtering in the spatial domain can be achieved by a beamforming technique that consists of synthesizing virtual microphones with adjustable characteristics (steering direction and directivity pattern) [2, 3].

1.3 Spatial filter bank in Ambisonics

Ambisonics is a recording and reproduction technique that can be used to create spatial sound for circular or spherical loudspeaker arrangements [4, 5]. The technique is based on the representation of the sound field as a combination of orthogonal basis functions, namely the spherical harmonics (SH). The spatial filtering operation (i.e. beamforming) can be elegantly and efficiently designed in the SH domain [6, 7], effectively decomposing an Ambisonics scene into different angular sectors.

Such partitioning is typically used for local *analysis* of the sound field, in particular to estimate the acoustic intensity or energy density [8, 9], to improve estimation techniques of directions of arrival [10], to analyze the directional properties of a reverberant field [11–16], to visualize activity maps [17], or to implement audio compression algorithms for bandwidth reduction [18]. More generally, spatial filter banks are potentially useful for *parametric* spatial processing methods [19], such as DirAC [20], Harpex [21], HO-DirAC [18, 22–24], Spare Recovery [25], or COMPASS [26].

The above-mentioned techniques are mainly concerned with the *analysis* or reproduction of the sound field. In the present study, we focus on the exploitation of Ambisonics filter banks for the creation of audio effects, i.e. for the *transformation* of the sound scene, typically for artistic production (or post-production) applications.

The principle of the approach is illustrated in Figure 1: the incoming Ambisonics stream x is decomposed into K angular sectors by a filter bank derived from [18, 27, 28], and which is similar to a plane-wave decomposition (PWD) of the sound field [29]. After processing with various monophonic effects, the output stream y is recomposed by re-encoding the K plane waves, in the K analyzed directions, in the Ambisonics domain.

1.4 Ambisonics FX

Transforming a sound field by applying FX to an Ambisonics stream is not a new approach. Various tools are already available to sound engineers and composers. Some of these FX are operated globally on the sound scene, by

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

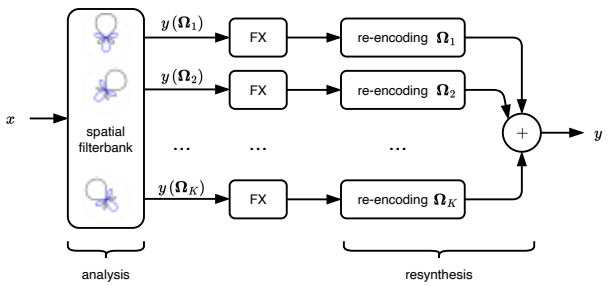


Figure 1. Proposed framework for the implementation of spatially localized audio FX.

application of a (frequency-independent) transformation matrix to the Ambisonics stream [30] [5, Chapter 5.2]; this includes in particular the rotation [4, 31], mirror [32], or spatial blur [33] effects. Other FX were first formulated in the spatial domain, then "translated" into the modal domain by discrete SH transform [34]; this methodology is fundamentally similar to the analysis/resynthesis filter bank shown in Figure 1. However, the filter bank approach may prove more intuitive for certain users, in particular sound engineers who easily understand the concept of virtual microphones. This strategy (of design in the spatial domain) notably makes it possible to design higher-order FX such as directional emphasis (or directional loudness) [34] and angular distortion (also known as dominance [32, 35] or warping [5, 36]). Another extension of these methods, based on PWD, was proposed by Hafsati et al. [37] in order to produce even more "fine" editing of the sound scene; the technique however relies on the assumption that the directions of S sources of interest are known (or have been previously estimated).

This article is organized as follows: in Section 2, we recall the theoretical background for the creation of a perfect reconstruction Ambisonics filter bank. In Section 3 we present the software tool `beamix` which implements such a filter bank. Consisting of an audio engine associated with a graphical user interface, the tool extends the transformation possibilities of the approaches cited above, while offering intuitive control.

2. BACKGROUND

This section briefly summarizes the spatial filter bank approach proposed by Hold et al. [18, 27, 28].

2.1 Axis-symmetric beamforming

Here we consider Ambisonics signals encoded on a basis of spherical harmonics $Y_n^m(\mathbf{\Omega})$, with N denoting the maximum order. $\mathbf{\Omega} \equiv (\theta,\ \phi)$ denotes the angular direction in the spherical coordinate system. The spherical harmonic addition theorem [7, Eq. 1.26] is written $\forall \mathbf{\Omega}, \mathbf{\Omega}' \in \mathbb{S}^2$:

$$\sum_{m=-n}^{n} \left[Y_n^m(\mathbf{\Omega})\right]^* Y_n^m(\mathbf{\Omega}') = \frac{2n+1}{\lambda^2}\,\mathcal{P}_n(\cos\Theta)\,, \quad (1)$$

where $\mathcal{P}_n$ is the Legendre polynomial of order n, and $\Theta = \angle(\mathbf{\Omega}, \mathbf{\Omega}')$ represents the angle between the directions $\mathbf{\Omega}$ and $\mathbf{\Omega}'$. λ here denotes a constant which depends only on the chosen variety of SH. In the case of N3D normalization, guaranteeing the orthonormality of the components [4, 38], we have $\lambda^2 = 1$; however most publications [5, 7, 28, 29, 39] adopt the convention $\lambda^2 = 4\pi$. For the sake of compatibility, we keep the generic notation λ in the formulas that follow.

It follows from the addition theorem that any axis-symmetric directional function, obtained by weighting the SH with $(N+1)$ real coefficients d_n, can be expressed in the form [40] [7, Eq. 5.24]:

$$y(\Theta) = \sum_{n=0}^{N} d_n\,\frac{2n+1}{\lambda^2}\,\mathcal{P}_n(\cos\Theta)\ . \quad (2)$$

This expression characterizes the output signal $y(\Theta)$ resulting from a modal beamforming operation.

2.2 Choice of spatial sampling

To create the spatial filter bank, we use K beamformers pointing in directions $\{\mathbf{\Omega}_k\}$ for $k = 1, ..., K$. These directions must be "uniformly" distributed, so as to ensure numerical quadrature over $\mathbb{S}^2$. Assuming a uniform distribution, the integration of a band-limited continuous function $f \in \mathbb{S}^2$ reduces to a discrete summation:

$$\int_{\mathbb{S}^2} f(\mathbf{\Omega})\,d\mathbf{\Omega} = \frac{\lambda^2}{K}\sum_{k=1}^{K} f(\mathbf{\Omega}_k)\ . \quad (3)$$

The spherical *t-design* sampling schemes [41, 42] are known to satisfy this quadrature condition (Eq. 3), provided that $t \geq (2N+1)$ [5, Section 4.7] [7, Section 3.4]. From now on, we assume that $\{\mathbf{\Omega}_k\}$ is such a spherical *t-design* arrangement. To evaluate the reconstruction capacity of the spatial filter bank, we are interested in the sum of the contributions of the K sectors, which we can write:

$$\begin{aligned}\sum_{k=1}^{K} y(\Theta_k) &= \sum_{k=1}^{K}\sum_{n=0}^{N} d_n\,\frac{2n+1}{\lambda^2}\,\mathcal{P}_n(\cos\Theta_k)\\ &= \sum_{k=1}^{K}\sum_{n=0}^{N} d_n \sum_{m=-n}^{n}\left[Y_n^m(\mathbf{\Omega})\right]^* Y_n^m(\mathbf{\Omega}_k)\\ &= \sum_{n=0}^{N} d_n \sum_{m=-n}^{n}\left[Y_n^m(\mathbf{\Omega})\right]^* \sum_{k=1}^{K} Y_n^m(\mathbf{\Omega}_k)\end{aligned} \quad (4)$$

(the spherical addition theorem Eq. 1 has been introduced between the first and second lines). Furthermore, by applying the quadrature equation Eq. 3 to the $Y_n^m(\mathbf{\Omega}_k)$ function, we have:

$$\sum_{k=1}^{K} Y_n^m(\mathbf{\Omega}_k) = \frac{K}{\lambda^2}\int_{\mathbb{S}^2} Y_n^m(\mathbf{\Omega})\,d\mathbf{\Omega}\,. \quad (5)$$

By virtue of the SH orthogonality, we know that:

$$\int_{\mathbb{S}^2} Y_n^m(\mathbf{\Omega})\,d\mathbf{\Omega} = \lambda\,\delta_{n0}\,, \quad (6)$$

where δ_{nm} represents the Kronecker symbol ($\delta_{nm} = 1$ if $n = m$, 0 otherwise). We can thus simplify the summation equation Eq. 4 over the K sectors:

$$\sum_{k=1}^{K} y\left(\Theta_k\right) = d_0 \left[Y_0^0\left(\mathbf{\Omega}\right)\right]^* \frac{K}{\lambda} = d_0 \frac{K}{\lambda^2}\,, \tag{7}$$

since $\forall \mathbf{\Omega} \in \mathbb{S}^2,\ Y_0^0\left(\mathbf{\Omega}\right) = \lambda^{-1}$, by construction.

2.3 Energy or amplitude compensation

As discussed in [18, 27, 28], it is necessary to introduce a scaling factor to ensure the perfect reconstruction of the filter bank. When amplitude reconstruction is needed, we introduce a factor β_A which must satisfy $\forall \mathbf{\Omega} \in \mathbb{S}^2,\ \sum_{k=1}^{K} \beta_A\ y\left(\mathbf{\Omega}_k\right) = 1$. Comparing this with Eq. 7, it follows that $\beta_A = \frac{\lambda^2}{d_0\,K}$ is a solution. When energy reconstruction is desired, we must introduce a factor β_E such that $\forall \mathbf{\Omega} \in \mathbb{S}^2,\ \sum_{k=1}^{K} \beta_E\ y^2\left(\mathbf{\Omega}_k\right) = 1$, and it is easily shown that $\beta_E = \frac{\lambda^2}{d_0^2\,K}$. These compensation factors are introduced into the resynthesis stage (Figure 1) in order to ensure the perfect reconstruction of the filter bank (see [18, Sections III.A.5 and III.A.6] for more details).

2.4 Choice of the beampattern

After choosing the directions $\left\{\mathbf{\Omega}_k\right\}$, one needs to determine the directivity pattern (i.e. the selectivity) of the beamformer. A natural choice is the hyper-cardioid beampattern of order N, as it performs a PWD of the sound field [7, 29]. It is realized using constant modal weights such that [40] [7, Eq. 6.10]:

$$\forall n \leq N,\ d_n = \frac{4\pi}{\left(N+1\right)^2}\,. \tag{8}$$

The resulting directivity pattern is also called a "regular beampattern" or "basic" (in Ambisonics vernacular), and it achieves the maximum directivity index [7, 40]. Despite this interesting property, the hyper-cardioid pattern can be criticized because it exhibits strong sidelobes (lateral and rear). This point is particularly sensitive when creating a spatial filter bank because the sidelobes of the K beams interfere. Therefore, other beampatterns may be preferred; among the different commonly used options, we can for example retain two of interest: (a) the cardioid pattern (also referred to as "in-phase" in Ambisonics dialect) which is designed to suppress signals arriving in the opposite direction ($\Theta = \pm 180°$) while providing an optimally flat response in that direction. The response of the cardioid beamformer satisfies [43, Section 6.1]:

$$y\left(\Theta\right) = \left(\frac{1+\cos\Theta}{2}\right)^N, \tag{9}$$

and it is achieved with the weights [4, Eq. 3.91]:

$$\forall n \leq N,\ d_n = 4\pi\,\frac{\left(N!\right)^2}{\left(N+n+1\right)!\,\left(N-n\right)!}\,. \tag{10}$$

(b) The so-called max-r_E pattern is also interesting because, as the name suggests, it is designed to maximize the norm r_E of the energy vector [4, 5]. The properties of this beampattern are somewhat similar to the super-cardioid pattern (providing the maximum front-to-back ratio). The max-r_E weighting is given by [4, Eq. 3.89]:

$$\forall n \leq N,\ d_n = \mathcal{P}_n\left(r_E(N)\right)\,, \tag{11}$$

where $r_E(N)$ is the largest root of $\mathcal{P}_{N+1}\left(\cdot\right)$; this can be approximated by $r_E(N) \approx \cos\left(\frac{137.9°}{N+1.52}\right)$ (see [5, Section A.3.7]).

3. IMPLEMENTATION

3.1 Description of the tool

The processing chain proposed in Figure 1 is implemented in the `C++` language in the form of two external objects for the Max environment [1]. The `spat5.hoa.beamix~` object renders the signal processing, and the `spat5.hoa.beamix` object is the associated graphical user interface (GUI), developed with the Juce framework [2]. The objects are integrated into the Spat library [44], distributed free of charge by Ircam [3].
The GUI is shown in Figure 5. The left part shows a map projection of the sphere $\mathbb{S}^2$; different types of projection are available, and by default the Mollweide projection is used. The meridians and parallels are displayed in light gray. The K angular sectors are represented (here in blue); their index is placed at the center of the area, and the sectors are delimited by the edges of the corresponding spherical Voronoi diagram. In the example of Figure 5, sector #17 (in red) is currently selected by the user. The right half of the interface shows the effects controllers for the selected area(s). Currently implemented are traditional audio effects, namely gain and delay adjustment, parametric EQ (with 8 adjustable stages) and compressor/expander. To complete this toolchain, it is also possible to load (from an audio file) a finite impulse response filter which is rendered by an optimized convolution engine. Finally it is possible to mute or solo the sector, which is typically useful when tweaking the FX settings. Vu-meters display the input (pre-FX) and output (post-FX) levels of the sector.
In the case of a 2D Ambisonics scene, a polar representation of the directional beams is displayed (Figure 2) instead of the map projection (Figure 5, left).

3.2 Adjustable beampattern

In Figure 2 (with $N = 3$, $K = 7$, and hyper-cardioid patterns), we clearly observe the role of the sidelobes which significantly "leak" into all other sectors. As mentioned in Section 2.4, this sometimes motivates the use of alternative beampatterns with less pronounced sidelobes (but therefore with lower selectivity), such as *in-phase* or max-r_E. In `spat5.hoa.beamix`, to facilitate user experience, the selectivity of the beamformer is adjustable with a continuous parameter α, allowing a tradeoff between cardioid ($\alpha = 0\%$), max-r_E ($\alpha = 50\%$) and hyper-cardioid

[1] https://cycling74.com
[2] https://juce.com
[3] https://forum.ircam.fr/projects/detail/spat/

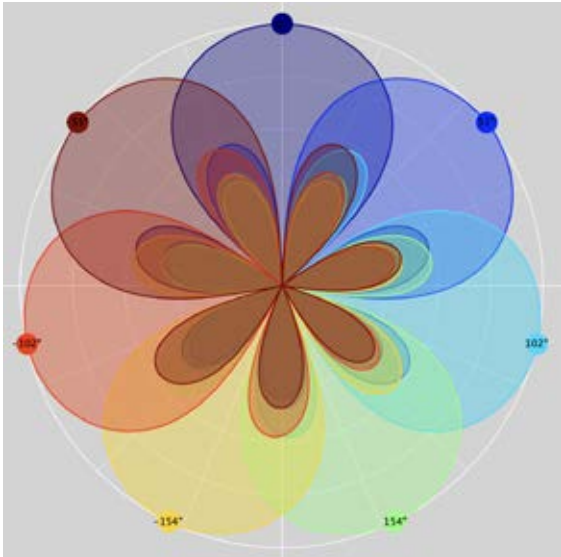


Figure 2. `spat5.hoa.beamix` filter bank (polar plot of hyper-cardioid beampatterns) in the 2D case. Here for $N = 3$, therefore $K = 7$ beams. The radial scale is logarithmic, with 6 dB per division.

($\alpha = 100\%$) beampatterns. The transition between these three diagrams is made by linear interpolation between the weights d_n, with an approach similar to [45]. Figure 3 illustrates the polar diagram of the beamformer for different values of α.

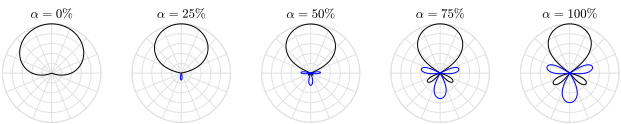


Figure 3. Directivity pattern with adjustable selectivity. $\alpha = 0\%$, $\alpha = 50\%$, and $\alpha = 100\%$ respectively correspond to a cardioid, max-r_E, and hyper-cardioid beampatterns. The figures are for the 3D case, with $N = 3$. The radial scale is logarithmic, with 6 dB per division.

3.3 UI and ergonomics

The two external objects `spat5.hoa.beamix~` and `spat5.hoa.beamix` benefit from the main ergonomic features introduced in version 5 of Spat [46] : (a) Communication is based on OSC messages and bundles offering inter-operability and flexibility of use; typically one can use pattern matching (for example the syntax `/beam/[10-20]`) simultaneously to send messages to several sectors; the *copy to neighbors* option (see ⑦ on Figure 5) also allows one to propagate the settings of a zone to all adjacent sectors; more generally, it is possible to copy/paste the parameters from one sector to another. (b) It is easy to export/import/interpolate presets. (c) The audio engine is compatible with multichannel patchcords "MC" (see Figure 4), and highly optimized and vectorized.
Nevertheless, user experience can still be improved; the two

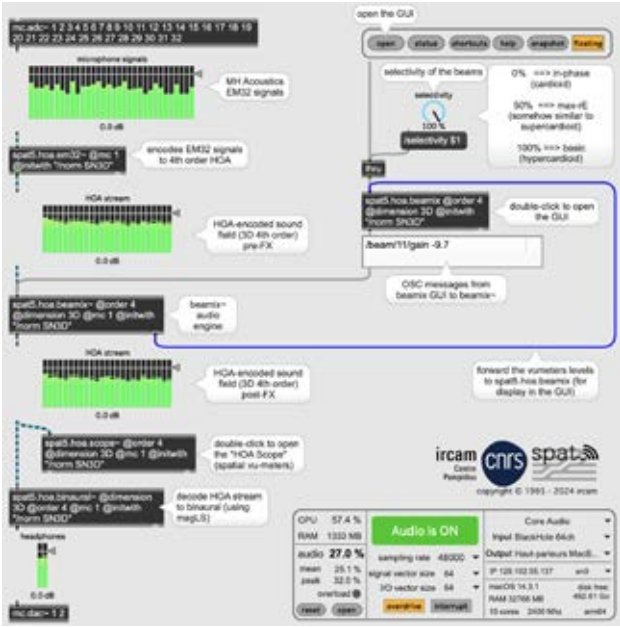


Figure 4. Example of a Max patch with `spat5.hoa.beamix~` processing a sound scene (here captured by an Ambisonics microphone, then rendered binaurally to headphones).

main challenges are due to the fact that (a) the tool operates in the space-time-frequency domain, where it is difficult to understand/visualize all dimensions simultaneously. (b) For higher orders N, the number of sectors is considerable and editing the scene quickly becomes tedious; it would therefore be useful to find efficient strategies for tweaking parameters either at the macro or micro level. There is no doubt that user feedback will help gradually to enrich the control interface and improve these aspects.

3.4 Discussion

The effects proposed in the processing chain (Figure 5, right panel) are (for the moment) quite basic; nevertheless, they offer a richer range of treatments than the other tools of the state of the art. In particular, non-linear processing (compressor/limiter) is very useful for correcting certain sound recordings, in order locally to emphasize sound elements that could be masked. Such non-linear FX cannot be applied directly in the SH domain because they would destroy the global spatial information; on the other hand, it is legitimate to use them locally on one or more sectors.
In future work, we would like to enrich the range of available FX. For this, a relevant solution would be to insert existing (monophonic) plugins effects in each sector. Technically, this solution can already be implemented using Max's `mc.vst~` object; however the operation is laborious, and native integration in `spat5.hoa.beamix~` could streamline the user workflow.
The presence of a (monophonic) convolution engine on each sector already offers certain possibilities, however limited to the synthesis of linear and invariant systems. One possible application is the production of an artificial

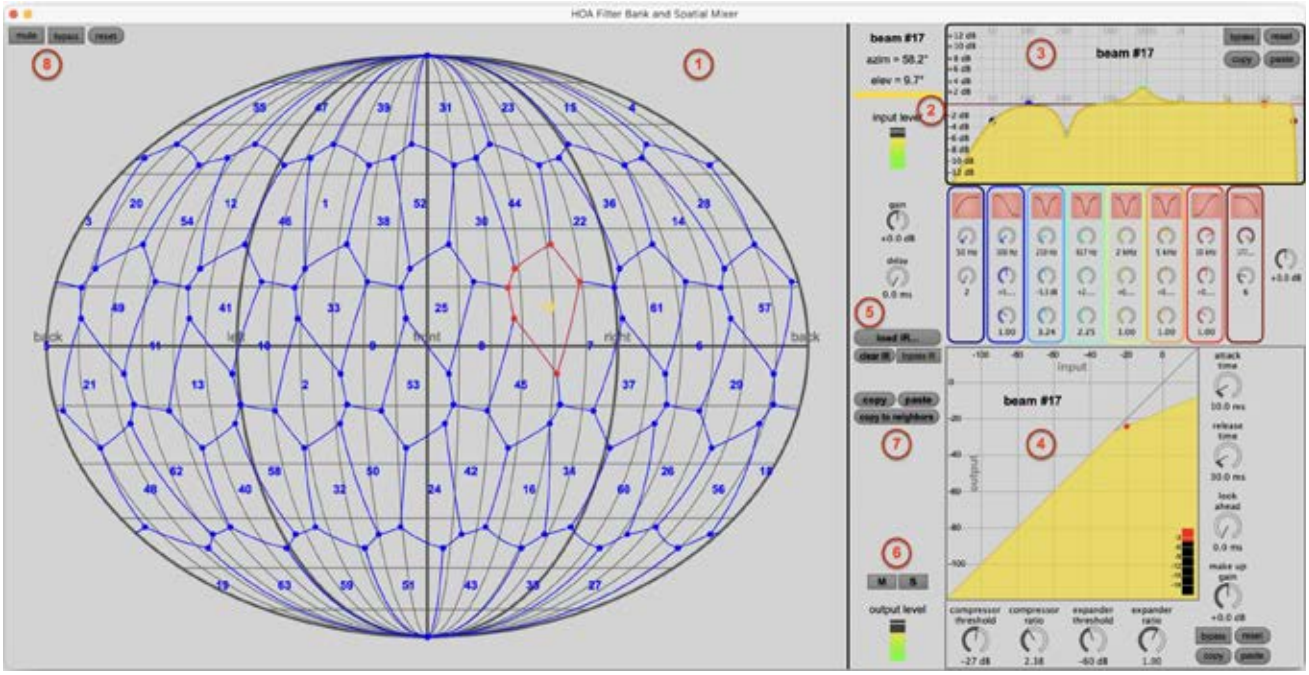


Figure 5. `spat5.hoa.beamix` GUI. Here in the 3D case, for order $N = 4$ and $K = 64$ beams. ① Cartographic projection of the sphere (here in Mollweide projection) and representation of the K angular sectors. ② Audio FX settings for the selected sector (here #17). ③ Equalizer for the selected sector. ④ Compressor and expander. ⑤ Button to load a finite impulse response filter. ⑥ Mute, solo, and output vu-meter for the selected sector. ⑦ Copy/paste the sector settings. ⑧ Global mute, bypass and reset controls (applied to all sectors).

reverberation effect whose characteristics (e.g. decay) vary depending on the directions [15]. However, it should be kept in mind that the calculation cost is likely to be considerable (in particular for "long" impulse responses) since it is proportional to the number K of sectors.

In addition to transforming sound scenes, another relevant use case concerns the editing of directional room impulse responses (DRIR) as captured by a spherical array of microphones. Once encoded in Ambisonics format, these responses can be edited locally, in the space-time-frequency domain, with `beamix`. This allows one, for example, to filter unwanted reflections, as already exemplified in [47]. It has also been shown that for DRIR exhibiting strong anisotropic characteristics, processing in the PWD can be more relevant than in the SH domain [12, 13]. The use of the Ambisonics filter bank is therefore appropriate for such responses. Finally, note that for editing DRIR, it is generally easier to work offline rather than in Max (or any other real-time oriented environment); this motivated the development of bindings of `beamix` for the command line (included in the Spat package) and for Matlab (currently not distributed publicly).

4. CONCLUSION

In this article we presented `beamix`, an audio engine and its graphical control interface allowing directional audio effects, i.e. FX localized in certain angular sectors. The effects are applied to an Ambisonics audio stream, previously decomposed by a state-of-the-art spatial filter bank. The latter performs beamforming with virtual microphones and is analogous to a plane-wave decomposition. This approach allows greater flexibility compared to transformations applied in the modal domain. We presented the underlying mathematical formalism, then detailed the implementation of the tool, in particular its graphical interface which is intended to be intuitive and flexible.
However, several avenues for future improvement are considered. To enrich the arsenal of available FX, it would be relevant to allow the insertion of existing plugins (VST or other formats) in the processing chain. In addition, the ergonomics of the user interface can still be improved, and, in particular, it remains to invent strategies for joint editing (or visualization) of several neighboring sectors. And, more fundamentally, it remains for us to explore new approaches for the even more generic design of spatial filter banks, particularly in the wavelet domain [48], as this might offer adaptive spatial resolution depending on the areas of interest.

5. REFERENCES

[1] U. Zölzer, *DAFX: Digital Audio Effects*, 2nd ed. John Wiley & Sons, March 2011. https://doi.org/10.1002/9781119991298

[2] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Springer, March 2008. https://doi.org/10.1007/978-3-540-78612-2

[3] J. Benesty, J. Chen, and C. Pan, *Fundamentals of Differential Beamforming*. Springer, 2016. https://doi.org/10.1007/978-981-10-1046-0

[4] J. Daniel, "Représentation de champs acoustiques, application à la transmission et à la reproduction de scènes sonores complexes dans un contexte multimédia," Ph.D. dissertation, Université de Paris VI, September 2000. https://www.theses.fr/2000PA066581

[5] F. Zotter and M. Frank, *Ambisonics: A Practical 3D Audio Theory for Recording, Studio Production, Sound Reinforcement, and Virtual Reality*, 1st ed. Cham, Switzerland: Springer, 2019. https://doi.org/10.1007/978-3-030-17207-7

[6] B. Rafaely, "Spatial Sampling and Beamforming for Spherical Microphone Arrays," in *Proc. of the Hands-Free Speech Communication and Microphone Arrays*, Trento, Italy, May 2008. https://doi.org/10.1109/hscma.2008.4538673

[7] ——, *Fundamentals of Spherical Array Processing*, 2nd ed. Cham, Switzerland: Springer, February 2019. https://doi.org/10.1007/978-3-319-99561-8

[8] A. Politis and V. Pulkki, "Acoustic intensity, energy-density and diffuseness estimation in a directionally-constrained region," *arXiv*, September 2016. https://doi.org/10.48550/arXiv.1609.03409

[9] L. McCormack, S. Delikaris-Manias, A. Politis, D. Pavlidi, A. Farina, D. Pinardi, and V. Pulkki, "Applications of Spatially Localized Active-Intensity Vectors for Sound-Field visualization," *Journal of the Audio Engineering Society*, vol. 67, no. 11, pp. 840 – 854, November 2019. https://doi.org/10.17743/jaes.2019.0041

[10] D. P. Jarrett, E. A. P. Habets, and P. A. Naylor, "3D source localization in the spherical harmonic domain using a pseudointensity vector," in *Proc. of the 18th European Signal Processing Conference (EUSIPCO)*, Aalborg, Denmark, August 2010, pp. 442 – 446.

[11] M. Nolan, E. Fernandez Grande, J. Brunskog, and C.-H. Jeong, "A wavenumber approach to quantifying the isotropy of the sound field in reverberant spaces," *Journal of the Acoustical Society of America*, vol. 143, no. 4, pp. 2514–2526, 2018. https://doi.org/10.1121/1.5032194

[12] P. Massé, T. Carpentier, O. Warusfel, and M. Noisternig, "Denoising Directional Room Impulse Responses with Spatially Anisotropic Late Reverberation Tails," *Applied Sciences*, vol. 10, no. 3, pp. 1 – 17, February 2020. https://doi.org/10.3390/app10031033

[13] ——, "A robust denoising process for spatial room impulse responses with diffuse reverberation tails," *Journal of the Acoustical Society of America*, vol. 147, no. 4, pp. 2250 – 2260, April 2020. https://doi.org/10.1121/10.0001070

[14] M. Berzborn and M. Vorländer, "Directional sound field decay analysis in performance spaces," *Building Acoustics*, vol. 28, no. 3, pp. 249 – 263, 2021. https://doi.org/10.1177/1351010X20984622

[15] B. Alary, P. Massé, S. J. Schlecht, M. Noisternig, and V. Välimäki, "Perceptual analysis of directional late reverberation," *Journal of the Acoustical Society of America*, vol. 149, no. 5, pp. 3189 – 3199, 2021. https://doi.org/10.1121/10.0004770

[16] C. Hold, T. McKenzie, G. Götz, S. J. Schlecht, and V. Pulkki, "Resynthesis of Spatial Room Impulse Response Tails With Anisotropic Multi-Slope Decays," *Journal of the Audio Engineering Society*, vol. 70, no. 6, pp. 526 – 538, June 2022. https://doi.org/10.17743/jaes.2022.0017

[17] L. McCormack and A. Politis, "SPARTA & COMPASS: Real-time implementations of linear and parametric spatial audio reproduction and processing methods," in *Proc. of the AES International Conference on Immersive and Interactive Audio*, York, UK, March 2019. http://www.aes.org/e-lib/browse.cfm?elib=20417

[18] C. Hold, V. Pulkki, A. Politis, and L. McCormack, "Compression of Higher-Order Ambisonic Signals Using Directional Audio Coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 651 – 665, 2024. https://doi.org/10.1109/taslp.2023.3328284

[19] V. Pulkki, S. Delikaris-Manias, and A. Politis, *Parametric Time-Frequency Domain Spatial Audio*, V. Pulkki, S. Delikaris-Manias, and A. Politis, Eds. Wiley, October 2017. https://doi.org/10.1002/9781119252634

[20] V. Pulkki, "Spatial Sound Reproduction with Directional Audio Coding," *Journal of the Audio Engineering Society*, vol. 55, no. 6, pp. 503 – 516, June 2007. http://www.aes.org/e-lib/browse.cfm?elib=14170

[21] S. Berge and N. Barrett, "High Angular Resolution Planewave Expansion," in *Proc. of the 2nd International Symposium on Ambisonics and Spherical Acoustics*, Paris, France, May 2010.

[22] A. Politis, J. Vilkamo, and V. Pulkki, "Sector-Based Parametric Sound Field Reproduction in the Spherical Harmonic Domain," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 5, pp. 852 – 866, August 2015. https://doi.org/10.1109/jstsp.2015.2415762

[23] V. Pulkki, A. Politis, G. Del Galdo, and A. Kuntz, "Parametric Spatial Audio Reproduction with Higher-Order B-Format Microphone Input," in *Proc. of the 134th Convention of the Audio Engineering Society (AES)*, Roma, Italy, May 2013. http://www.aes.org/e-lib/browse.cfm?elib=16820

[24] C. Hold, L. McCormack, A. Politis, and V. Pulkki, "Optimizing Higher-Order Directional Audio Coding with Adaptive Mixing and Energy Matching for

Ambisonic Compression and Upmixing," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, October 2023. https://doi.org/10.1109/waspaa58266.2023.10248179

[25] A. Wabnitz, N. Epain, A. McEwan, and C. Jin, "Upscaling Ambisonic sound scenes using compressed sensing techniques," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, October 2011. https://doi.org/10.1109/aspaa.2011.6082301

[26] A. Politis, S. Tervo, and V. Pulkki, "COMPASS: Coding and Multidirectional Parameterization of Ambisonic Sound Scenes," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Calgary, AB, Canada, April 2018, pp. 6802 – 6806. https://doi.org/10.1109/icassp.2018.8462608

[27] C. Hold, S. J. Schlecht, A. Politis, and V. Pulkki, "Spatial Filter Bank in the Spherical Harmonic Domain: Reconstruction and Application," in *Proc. of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, NY, USA, October 2021. https://doi.org/10.1109/waspaa52581.2021.9632709

[28] C. Hold, A. Politis, L. McCormack, and V. Pulkki, "Spatial Filter Bank Design in the Spherical Harmonic Domain," in *Proc. of the European Signal Processing Conference (EUSIPCO)*, Dublin, Ireland, August 2021. https://doi.org/10.23919/EUSIPCO54536.2021.9616091

[29] B. Rafaely, "Plane-wave decomposition of the sound field on a sphere by spherical convolution," *Journal of the Acoustical Society of America*, vol. 116, no. 4, pp. 2149 – 2157, October 2004. https://doi.org/10.1121/1.1792643

[30] M. Chapman and P. Cotterell, "Towards a comprehensive account of valid ambisonic transformations," in *Proc. of the 1st Ambisonics Symposium*, Graz, Austria, 2009.

[31] J. Daniel, "Evolving Views on HOA : From Technological To Pragmatic Concerns," in *Proc. of the 1st Ambisonics Symposium*, Graz, Austria, June 2009.

[32] M. Chapman, "Symmetries of Spherical Harmonics: Applications to Ambisonics," in *Proc. of the 1st Ambisonics Symposium*, Graz, Austria, August 2009.

[33] T. Carpentier, "Ambisonic spatial blur," in *Proc. of the 142nd Convention of the Audio Engineering Society (AES)*, Berlin, Germany, May 2017. https://www.aes.org/e-lib/browse.cfm?elib=18606

[34] M. Kronlachner and F. Zotter, "Spatial transformations for the enhancement of Ambisonic recordings," in *Proc. of the 2nd International Conference on Spatial Audio (ICSA)*, Erlangen, Germany, February 2014.

[35] M. A. Gerzon and G. J. Barton, "Ambisonic Decoders for HDTV," in *Proc. of the 92nd Convention of the Audio Engineering Society (AES)*, Vienna, Austria, March 1992. https://www.aes.org/e-lib/browse.cfm?elib=6788

[36] H. Pomberger and F. Zotter, "Warping of 3D ambisonic recordings," in *Proc. of the Ambisonics Symposium*, Lexington, KY, USA, June 2011.

[37] M. Hafsati, N. Epain, and J. Daniel, "Editing Ambisonic Sound Scenes," in *Proc. of the 4th International Conference on Spatial Audio (ICSA)*, Graz, Austria, September 2017.

[38] T. Carpentier, "Normalization schemes in Ambisonic: does it matter?" in *Proc. of the 142nd Convention of the Audio Engineering Society (AES)*, Berlin, Germany, May 2017.

[39] N. A. Gumerov and R. Duraiswami, *Fast Multipole Methods for the Helmholtz Equation in Three Dimensions*, 1st ed. Elsevier, 2005. https://doi.org/10.1016/b978-0-08-044371-3.x5000-5

[40] J. Meyer and G. W. Elko, "Spherical microphone arrays for 3d sound recording," in *Audio Signal Processing for Next-Generation Multimedia Communication Systems*, Y. Huang and J. Benesty, Eds. Kluwer, 2004, ch. 3, pp. 67 – 89. https://doi.org/10.1007/1-4020-7769-6_3

[41] R. H. Hardin and N. J. A. Sloane, "McLaren's Improved Snub Cube and Other New Spherical Designs in Three Dimensions," *Discrete & Computational Geometry*, vol. 15, no. 4, pp. 429 – 441, 1996. https://doi.org/10.1007/bf02711518

[42] M. Gräf and D. Potts, "On the computation of spherical designs by a new optimization approach based on fast spherical Fourier transforms," *Numerische Mathematik*, vol. 119, no. 4, pp. 699 – 724, 2011. https://doi.org/10.1007/s00211-011-0399-7

[43] T. Carpentier, "Spherical beampatterns with fractional orders," in *Proc. of the Forum Acusticum, 10th Convention of the European Acoustics Association (EAA)*, Torino, Italy, September 2023, pp. 607 – 614. https://doi.org/10.61782/fa.2023.0531

[44] ——, "Spat: a comprehensive toolbox for sound spatialization in Max," *Ideas Sónicas*, vol. 13, no. 24, pp. 12 – 23, June 2021. https://hal.science/hal-03356292

[45] T. Carpentier, O. Warusfel, and J.-M. Jot, "Software Tools for Flexible Control of Radiation Synthesis," in *Proc. of the 2nd International Conference on Immersive and 3D Audio (I3DA)*, Bologna, Italy, September 2023. https://doi.org/10.1109/I3DA57090.2023.10289260

[46] T. Carpentier, "A new implementation of Spat in Max," in *Proc. of the 15th Sound and Music Computing Conference (SMC)*, Limassol, Cyprus, July 2018, pp. 184 – 191. https://doi.org/10.5281/zenodo.1422552

[47] T. Carpentier, T. Szpruch, M. Noisternig, and O. Warusfel, "Parametric control of convolution based room simulators," in *Proc. of the International Symposium on Room Acoustics (ISRA)*, Toronto, Canada, June 2013. https://hal.science/hal-01106836

[48] B. Yeo, W. Ou, and P. Golland, "On the Construction of Invertible Filter Banks on the 2-Sphere," *IEEE Transactions on Image Processing*, vol. 17, no. 3, pp. 283 – 300, March 2008. https://doi.org/10.1109/tip.2007.915550

TOWARD A NOVEL SET OF PINNA ANTHROPOMETRIC FEATURES FOR INDIVIDUALIZING HEAD-RELATED TRANSFER FUNCTIONS

Davide FANTINI (davide.fantini@unimi.it) (0000-0003-1332-0890),
Stavros NTALAMPIRAS (stavros.ntalampiras@unimi.it) (0000-0003-3482-9215),
Giorgio PRESTI (giorgio.presti@unimi.it) (0000-0001-7643-9915), and
Federico AVANZINI (federico.avanzini@unimi.it) (0000-0002-1257-5878)

Laboratory of Music Informatics (LIM), Department of Computer Science, **University of Milan**, Italy

ABSTRACT

Head-related transfer functions (HRTFs) are essential for spatial audio rendering. However, to provide a proper individual experience, the HRTF should be compliant with the anatomical traits of the listener. Therefore, estimating the individual HRTF based on anthropometric features is a widespread and promising approach. In this work, we propose an extension of the pinna anthropometric parameters commonly used in the related literature. The objective is to provide a more detailed description of the pinna for the purpose of individualizing HRTFs. The entirety of these parameters was extracted using a set of landmarks that were manually annotated on the shape of the pinna. The proposed parameters were compared to the classical ones for selecting the best-match HRTF in the median plane via anthropometry matching. Prior to HRTF selection, we devised a procedure to retain only the relevant anthropometric parameters. The obtained results were evaluated using both objective metrics and auditory model evaluation indicating a significant, though limited, improvement when using the proposed parameters.

1. INTRODUCTION

The direction-dependent effect of human body on sound waves is interpreted by a human listener as spatial cues. Head-related transfer functions (HRTFs) [1] describe this effect as a linear time-invariant (LTI) system. An HRTF set is a collection of transfer functions, one for each source position of interest and for each ear. HRTF sets encode the cues for spatial hearing, such as interaural time and level differences (ITD, ILD) and monaural spectral modifications. Each person has an individual HRTF as these cues depend on the body anatomy. In this respect, the pinna is of particular interest as it is responsible for key monaural spectral cues. The pinna influence on HRTF can be isolated by extracting the pinna-related transfer function (PRTF).

HRTFs find application in multiple fields [2] and are essential in extended reality (XR) frameworks where spatial audio is involved. However, to provide an immersive auditory experience, HRTFs should be compliant with the user's anatomical traits. In end-user applications, a non-individual HRTF (e.g., recorded from a dummy head) is often employed, potentially causing inadequacies such as front-back and up-down confusion, and degradation in elevation localization [3–5]. On the other hand, the HRTF acoustic measurement entails high-cost equipment and time-consuming sessions. For these reasons, the literature includes several works on HRTF individualization, i.e., the estimation of individual HRTF without the need for acoustic measurement. Several approaches for HRTF individualization are based on anthropometric parameters [2, 6, 7]. The CIPIC anthropometric specification [8] is the prevalent one, despite evidence of its incompleteness [9]. Some variations to the CIPIC pinna parameters have been proposed in the literature. However, few attempts have been made to evaluate the relevance of the proposed variations.

In this work, we propose some new pinna parameters in addition to those used in the HUTUBS dataset [10], which are derived from CIPIC. The new parameters are intended to describe in more detail the pinna anatomy for the purpose of individualizing PRTF. In this regard, we especially focused on the concha and fossa triangularis given their influence on HRTF [9, 11], and we considered measurement types different from the classical distances and angles. The effectiveness of new parameters was evaluated for selecting the best-match PRTF via anthropometry matching, in comparison to the sole HUTUBS set. Prior to PRTF selection, an anthropometric selection step was performed to retain only the relevant parameters. The PRTF selection results were evaluated using both spectral distortion (SD) and localization metrics obtained with an auditory model.

The Matlab code to extract the pinna anthropometric parameters described in this paper is publicly available [1]. Also, the data generated in our research as well as a detailed documentation of the pinna parameters are included in supplementary materials [2].

2. BACKGROUND

HRTF individualization is often based on anthropometric parameters [2, 6, 7] due to the relationship between HRTF and body anatomy. These parameters are usually defined

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] https://github.com/DavideFantini/pinna-anthropometry-extraction
[2] https://doi.org/10.5281/zenodo.10805884

as distances between specific body points, although angles are also used. Several anthropometric specifications have been proposed in the HRTF field, usually defined as two-dimensional sketches [1, Ch. 7]. Early anthropometric specifications include the one used for the KEMAR mannequin's design [12] and the one proposed by Middlebrooks [13]. In 2001, Algazi et al. [8] proposed the CIPIC specification, which includes ten parameters for the pinna and 17 for the head and torso. This specification is still widely used in HRTF datasets and HRTF individualization, although some variations have been proposed. For example, in the HUTUBS dataset, CIPIC parameters, with some variations in their definitions, were reported along with two new pinna parameters for the cavum concha. In the CHEDAR dataset [14], two pinna parameters were proposed for the crus helicis and the parabola effect of the pinna, in addition to five CIPIC parameters. Some of the anthropometric parameters defined in the standard GB/T 2428-1998 [15] were used for HRTF individualization [16] and HRTF datasets [17] in addition to the CIPIC ones. Recently, Iida et al. [18] measured ten distances between the tragus and other points on the pinna, while Stitt and Katz [9] defined a set of control points to measure CIPIC pinna parameters and other new ones. Teng and Zhong [19] proposed five area parameters of different pinna parts. However, the authors did not provide clear definitions for them. Despite the proposed CIPIC variations, currently, there is no complete and inter-independent anthropometric specification for HRTF.

The pinna anatomy is of particular interest in the HRTF field. Several studies have demonstrated its influence on vertical localization and front-back disambiguation [11, 20–23]. The role of pinna cavities has been analyzed with respect to the HRTF magnitude spectrum [20] and the localization in the median plane [23], while some studies focused on anthropometry [24, 25]. Literature findings suggest the influence of concha and triangular fossa on HRTF. In particular, Takemoto et al. [11] demonstrated that the nodes and anti-nodes occurring in these two pinna cavities are related to the notches of the median plane PRTF. Stitt and Katz [9] confirmed these findings by varying anthropometry to deform a 3D pinna model and comparing the corresponding HRTFs obtained via numerical simulation. The authors found a significant impact of concha and fossa triangularis parameters. Such findings suggest the incompleteness of widely used anthropometric specifications, such as CIPIC. For instance, parameters for the fossa triangularis are rare in these specifications.

3. PINNA PARAMETERS

3.1 Landmarks annotation scheme

In the following, we refer to the horizontal and vertical dimensions of depth images as x and y coordinates, respectively, while the z coordinate refers to depth. The proposed pinna parameters are extracted using a set of landmarks that are manually annotated on pinna depth images. Fig. 1 shows the scheme followed to annotate the landmarks, which comprises a set of points placed on the pinna to describe its shape. This scheme is a revised version of the one previously proposed by us [26]. The scheme includes $K = 205$ landmarks, providing a fine resolution of the pinna shape. This scheme divides the pinna shape into four parts: outer helix, concha, fossa triangularis, and inner helix. Fig. 1 shows the starting and ending landmarks indices for each part suggesting the landmarks annotation order. The first 49 landmarks (blue in Fig. 1) describe the outer helix edge starting from the attachment of the helix to the head (superior part of the pinna) and following the helix downwards to the inferior lobulus extremity. Other 68 landmarks (purple in Fig. 1) describe the concha starting from the inferior part of the crus helicis attachment to the cavum conchae, and following the entire concha edge ending under the inner helix edge. Other 18 landmarks (orange in Fig. 1) describe the fossa triangularis from the attachment of the antihelix (crus antihelicis superior) below the helix, to the point where the fossa triangularis edge reaches the border with the cymba conchae. The last 70 landmarks (yellow in Fig. 1) describe the inner helix, starting from the superior edge of the crus helicis attachment to the cymba conchae, and following the helix upward to its superior extremity, then continuing downward.

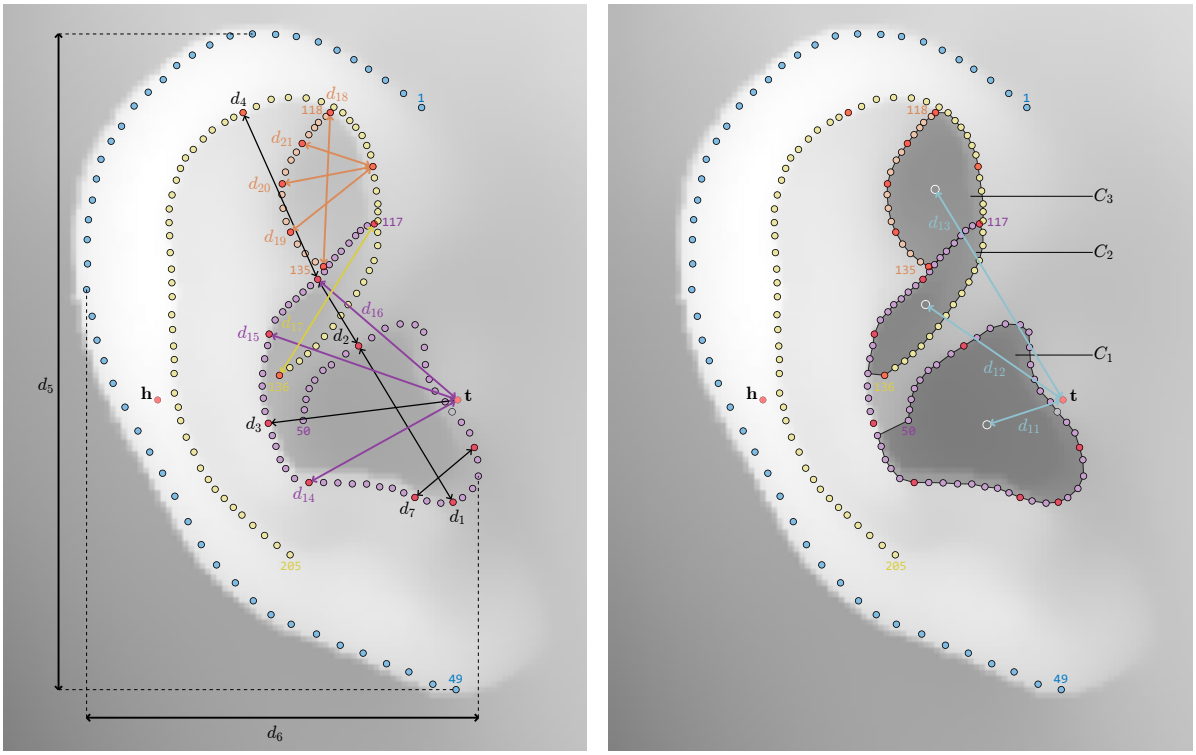
Landmarks are evenly spaced and placed in coherent positions across the pinnae. To ensure this coherence during annotation, we adjusted the landmark of a template shape superimposed on the image to match the current pinna. The landmarks serve to measure anthropometric parameters, therefore only the main contours of the pinna shape involved in the measurements are covered. Landmarks are placed near the border of the pinna contours, but not on the exact edge, to minimize the risk of having a landmark with a significantly incorrect z value. For instance, a slight misplacement of a landmark on the outer helix can result in the landmark being positioned on the head.

3.2 Pinna anthropometry measurement

We developed automated procedures to extract a set $\mathcal{P}$ of 48 pinna anthropometric parameters using the annotated landmarks and pinna depth images. For the sake of conciseness, we provide here a brief, though complete, overview of the parameters. For details on their definitions and measurements, please refer to the anthropometry documentation provided as supplementary material [2].

Most of the parameters are measured using the 2D landmarks (x and y coordinates), while the remaining ones consider the depth information (z coordinate) as well. A subset of $\mathcal{P}$ replicates the pinna parameters defined in the HUTUBS specification. Additionally, we defined new parameters to thoroughly describe the pinna cavities relevant for HRTF, such as the fossa triangularis. We have also explored alternative measurements to the traditional distances and angles, including areas, volumes, and depths. Fig. 1a shows some of the HUTUBS and the proposed pinna distance parameters, while Fig. 1b shows the pinna cavities and their respective distances.

The position $\mathbf{t}$ of the tip of the tragus is used to measure some parameters. Therefore, we estimated $\mathbf{t}$ by selecting the point with the maximum z value in a small area around



(a) (b)

Figure 1. Example of the scheme for manually annotating the pinna shape. Landmarks are color-coded based on the pinna shape part: outer helix edge (blue), concha (purple), fossa triangularis (orange), and inner helix edge (yellow). Numbers on the first and last landmarks of each part suggest the annotation order. The grayscale colormap represents the depth (z) values. (a) Comparison between some of the HUTUBS (black arrows) and the proposed distance parameters represented by arrows (colored as the pinna shape part) between red landmarks. (b) Pinna cavities C_i (shaded areas) along with the distance parameters (light blue arrows) between the tragus $\mathbf{t}$ and the cavities centroids (white empty circles).

a predefined landmark (gray landmark in Fig. 1). Further, we estimated the helix position $\mathbf{h}$ by identifying the point with the maximum z value at the same y coordinate as $\mathbf{t}$.

3.2.1 *HUTUBS distance parameters*

In this paper, we considered the HUTUBS distance parameters d_h with $h = \{1, \ldots, 9\}$ (we neglected d_{10} because we were unable to replicate its measurement). The parameters $d_{1\text{–}4}$, and d_7 are defined as distances between specific pinna points. To measure them, we selected the pairs of landmarks that best approximate the points indicated in the HUTUBS definition (see Fig. 1a) and computed their Euclidean distance in x and y coordinates. The parameters d_5 and d_6, i.e., the pinna height and width, were measured as the landmarks range along the y and x coordinates, respectively. This slightly differs from the definition of d_6 in HUTUBS, but it allowed us to obtain a more robust measurement. An ad hoc procedure was developed for d_8 and d_9. To measure d_9, a horizontal slice of the pinna depth image is taken at the y coordinate of $\mathbf{t}$. Then, d_9 is set as the distance in x and z coordinates between $\mathbf{t}$ and the corner formed by the concha surface and the antihelix wall. The parameter d_8 is measured as the 3D distance between $\mathbf{t}$ and the deepest point in the incisura intertragica, i.e., the region enclosed by tragus and antitragus.

Table 1 shows the absolute difference between the anthropometric values $\mathbf{A}^h$ measured by us and those reported in the HUTUBS dataset. It should be noted that our goal was not to exactly replicate the HUTUBS parameters. However, the obtained errors can be considered sufficient and comparable to those reported in similar studies [26–28].

3.2.2 *Additional distance parameters*

In addition to the HUTUBS parameters, we propose eight new parameters $d_{14\text{–}21}$ (see Fig. 1a) defined as distances in x and y coordinates between specific pairs of landmarks. The parameters $d_{14\text{–}16}$ integrate the HUTUBS parameters d_1, d_3 and d_7 in describing the concha. They are defined as distances between $\mathbf{t}$ and characteristic points on the concha edge. The parameter d_{17} quantifies the length of the cymba conchae and is therefore complementary to d_2. The parameters $d_{18\text{–}21}$ characterize the shape of the fossa triangularis.

3.2.3 *Angle parameters*

In addition to distances, HUTUBS provides two angle parameters: the pinna rotation angle θ_1 and the pinna flare angle θ_2. We measured θ_1 as the angle between the y axis and the line passing through the pinna extremities in y coordinates represented by the same landmarks used to measure d_5. Then, θ_2 was measured as the angle in x and z coordinates between the x axis and the line passing through $\mathbf{t}$ and $\mathbf{h}$. In addition to HUTUBS angles, we

Parameter	Mean ± St. dev. (%)
d_1 [mm]	1.40 ± 1.03 (7.8)
d_2 [mm]	1.65 ± 1.15 (16.4)
d_3 [mm]	1.31 ± 0.99 (7.6)
d_4 [mm]	4.89 ± 2.23 (23.6)
d_5 [mm]	1.83 ± 1.45 (3.0)
d_6 [mm]	3.72 ± 2.59 (12.6)
d_7 [mm]	0.82 ± 0.69 (13.2)
d_8 [mm]	2.89 ± 1.69 (25.4)
d_9 [mm]	2.08 ± 1.38 (17.4)
θ_1 [°]	1.98 ± 1.37 (17.5)
θ_2 [°]	6.15 ± 3.87 (23.9)

Table 1. Mean and standard deviation of the absolute differences between the anthropometric values $\mathbf{A}^h$ measured by us and the values reported in the HUTUBS dataset. The mean error as a percentage of the HUTUBS parameter mean is reported between brackets.

defined the *pinna upper roll* θ_3 as the angle in y and z coordinates between $\mathbf{t}$ and the landmark with the maximum y value. Also, for each distance parameter d_n with $n = \{1, \ldots, 21\} \setminus \{5, 6, 8, 9\}$, we defined the corresponding angle r_n as the angle between the y axis and the line passing through the two measurement landmarks of d_n. These angle parameters quantify the relative alignment of each pair of pinna points in addition to their distance d_n.

3.2.4 Pinna cavities parameters

In the pinna shape, we defined the three regions C_1, C_2 and C_3 corresponding to the pinna cavities cavum conchae, cymba conchae and fossa triangularis, respectively. These regions can be easily delimited by selecting specific landmarks as their position on the pinna shape is known. We defined three additional parameters, namely d_{11}, d_{12} and d_{13}, as the distances in x and y coordinates between $\mathbf{t}$ and the centroids of C_1, C_2 and C_3, respectively. These parameters measure the relative position of the cavities in the pinna shape, which is a complementary information to the shape and size considered by other parameters. Fig. 1b shows the regions $C_{11\text{–}13}$ and the parameters $d_{11\text{–}13}$.

Further, the extracted parameters include the area a_i of each region C_i and the volume v_i of the corresponding polyhedron Q_i. To obtain Q_i, we joined the vertices of C_i in 3D coordinates with the depth image pixels enclosed by C_i and computed the 3D convex hull. Also, we measured the depth h_i of each pinna cavity C_i as the range in the z coordinate of the Q_i vertices.

4. PRTF SELECTION

This section describes an HRTF individualization approach based on the anthropometric values $\mathbf{A}$ extracted as discussed previously. This approach includes the preprocessing of $\mathbf{A}$ and HRTFs, the anthropometric parameters selection, and the PRTF selection via anthropometric matching.

4.1 Preprocessing

4.1.1 Pinna parameters

The anthropometric values $\mathbf{A}$ were preprocessed by centering each parameter to have a median of 0, and scaling to have an interquartile range of 1. This normalization technique ensures that the values of distinct parameters fall within a similar range, making their comparison unbiased by their original scales.

4.1.2 HRTF

The preprocessing of HRTFs aims to isolate the influence of the pinna, which is the focus of our work. First, we extracted the PRTF by applying a 1 ms Hann window to the HRIR starting from its onset [22]. The window eliminates the torso and shoulders effects while retaining the pinna one. Next, we extracted the directional transfer function (DTF) magnitude from the PRTFs [29]. The analysis was restricted to the 3–15 kHz frequency range, as the pinna influence above this range is marginal, and below it, the torso and shoulders effects are prevalent [1, 20–22, 25].

Further, the analysis was limited to the median plane, as in related works that focus on the pinna [26, 30–32]. The median plane can be considered a case study for the monaural spectral cues caused by the pinna, which are crucial for elevation perception [11, 21–23]. The findings on the median plane can be extended to other vertical planes by modeling the head and torso with other approaches [33, 34]. In the median plane, we considered the elevations between $-80°$ and $260°$ and discarded the angle $-90°$ (below the subject).

4.2 Anthropometric parameters selection

A selection of the parameters in the set $\mathcal{P}$ is conducted to keep only those relevant to PRTF. Candidates subsets $\mathcal{P}' \subseteq \mathcal{P}$ are compared to find the best set $\mathcal{P}^*$. This comparison is based on spectral distortion (SD). The SD between two PRTFs $\mathbf{H}_1$ and $\mathbf{H}_2$ at elevation φ is so defined [1, Sec. 5.1]:

$$SD^{\varphi}(\mathbf{H}_1, \mathbf{H}_2) = \sqrt{\frac{1}{|F|} \sum_{f \in F} \left(20 \log \frac{|\mathbf{H}_1(f, \varphi)|}{|\mathbf{H}_2(f, \varphi)|} \right)^2} [dB], \tag{1}$$

where F is the set of considered frequency bins. The mean $\bar{SD}$ is obtained by averaging SD^{φ} values of all elevations φ in the considered set Φ. For each pinna t, we computed the distance D between its anthropometry $\mathbf{A}_t$ with the anthropometry $\mathbf{A}_{t'}$ of every other pinna t' in the dataset:

$$D(t, t') = \sqrt{\sum_{p \in \mathcal{P}'} (\mathbf{A}_t^p - \mathbf{A}_{t'}^p)^2}. \tag{2}$$

The pinna t^* minimizing D is selected as the best-match given the candidate set $\mathcal{P}'$. We prevented the right pinna of a subject from being selected as best-match for the left pinna of the same subject, and vice versa. Then we computed the mean of the $\bar{SD}$ values obtained for each pair of PRTFs $\mathbf{H}_t$ and the corresponding best-match $\mathbf{H}_{t^*}$.

The candidate sets of parameters $\mathcal{P}'$ was picked as follows. Starting from $|\mathcal{P}|$ parameters, we iteratively decreased the number m of considered parameters. For each iteration, we evaluate all the combinations of m parameters taken from $\mathcal{P}^*_{m+1}$, i.e., the best set found in the previous iteration. The best set $\mathcal{P}^*_m$ with m parameters is selected as the one minimizing $\bar{SD}$. The search stops if there is no combination of m parameters reducing SD compared to the previous iteration. This procedure permits to evaluate fewer candidates $\mathcal{P}'$ than all possible $2^{|\mathcal{P}|} - 1$ combinations, which is computationally intensive.

4.3 PRTF selection procedure

Selection-based methods for HRTF individualization involve choosing the so-called best-matching HRTF based on the similarity between the test subject and those in a dataset. Similarity can be computed in various ways. In this work, we used a baseline approach commonly found in the literature, i.e., anthropometry matching [16, 28, 33, 35]. For each test pinna t, we computed the Euclidean distance $D(t, t')$ (see Eq. (2)) between its anthropometry $\mathbf{A}_t$ and the anthropometry $\mathbf{A}_{t'}$ of every other pinna t' in the dataset considering the best set of parameters $\mathcal{P}^*$. The PRTF of pinna t^* that minimizes D is selected as the best matched. The selection procedure considers the two pinnae of the subject as separate instances. Therefore, the PRTFs of two different subjects can be selected for the left and right pinnae of the same test subject.

5. RESULTS

5.1 Validation procedure

The PRTF selection approach was evaluated using the HUTUBS dataset, which includes 3D head meshes, anthropometry, and HRTFs measured for 58 subjects. We excluded the FABIAN dummy head and repeated measurements of the same subject, resulting in $N = 55$ subjects for validation. We extracted the pinna depth images from the HUTUBS 3D head meshes. For each pinna, we projected the 3D point cloud from the head lateral view onto a grid representing the depth image. This operation is straightforward as the head meshes are coherently placed in the coordinate system. The extracted depth images have a resolution of 140×160 pixels.

We used leave-one-subject-out cross-validation (LOOCV) to evaluate the PRTF selection procedure. Thus, we performed N splits of the dataset into training and test sets, even if each pinna is a separate instance. In the n-th split, both pinnae of subject n are in the test set, while the remaining subjects are in the training set. This prevented to select the left pinna in the training set for the right pinna of the same subject in the test set, and vice versa. To avoid biases, we performed all statistical measurements of the preprocessing and parameters selection procedures on the training set only. The outcomes were then applied to both the training and test sets.

For each pinna, we obtained PRTFs with $|\Phi| = 33$ elevations and $|F| = 71$ frequency bins by preprocessing the HRTFs from HUTUBS (see Sec. 4.1). In the evaluation, we compared the individual PRTFs $\mathbf{H}$ with two conditions $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$. The PRTFs $\hat{\mathbf{H}}^h$ are selected using the anthropometry $\mathbf{A}^h$ including the eleven HUTUBS pinna parameters $d_{1\text{–}9}$, θ_1 and θ_2. The PRTFs $\hat{\mathbf{H}}$ are selected using the anthropometry $\mathbf{A}$ including all 48 parameters discussed in this paper. For a fair comparison, we relied exclusively on the anthropometry extracted by us as described in Sec. 3.2, and we ignored the values reported in HUTUBS. Further, we used the left ear PRTF $\mathbf{H}^f$ measured from the FABIAN dummy head as a baseline condition.

5.2 Anthropometric parameters selection

Averaging the LOOCV results of anthropometric parameters selection, for the conditions $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$ we selected 8.7 and 43.6 parameters with a minimum $\bar{SD}$ of 5.44 and 5.32 dB, respectively. Fig. 2 shows the selection frequency of each parameter for the PRTFs $\hat{\mathbf{H}}$ as a percentage. Several parameters have been selected at every LOOCV iteration, such as d_7, d_{12}, d_{13}, d_{17}, θ_3, a_3, h_1, r_2, r_4, and $r_{19\text{–}21}$. Conversely, d_1 is quite often discarded. Other less frequently selected parameters are a_2, r_1, r_3, and r_{14}. There is no noticeable effect of the type of measurement on the selection frequency.

5.3 Objective evaluation

We computed the SD (see Eq. (1)) with the individual PRTFs $\mathbf{H}$ to objectively evaluate the two conditions $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$. Averaging the LOOCV results, we obtained the mean SD values 5.56 ± 1.93 dB and 5.34 ± 1.81 dB for $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$, respectively. As a reference, we also computed the mean SD between $\mathbf{H}$ and the FABIAN PRTF $\mathbf{H}^f$ which is 5.64 ± 2.7 dB. Despite the obtained SD means are close, statistical analysis [3] demonstrated that SD for $\hat{\mathbf{H}}$ is significantly improved compared to $\hat{\mathbf{H}}^h$ and $\mathbf{H}^f$ ($p < 0.001$), while there is no significant difference between $\hat{\mathbf{H}}^h$ and $\mathbf{H}^f$. This result suggests that the proposed parameters provide a significant, though limited, improvement of SD compared to the sole HUTUBS ones. To quantify the extent of this improvement, we computed Cohen's d effect size between $\hat{\mathbf{H}}$ and $\hat{\mathbf{H}}^h$. The obtained value of 0.35 is considered an effect size between small and medium [36].

Fig. 3 shows the elevation-dependent mean SD values for $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$. The lowest SD values around 3 dB are found above the head, while SD increases up to 9 dB for lower elevations. However, $\hat{\mathbf{H}}$ seems to provide the greatest improvement for these elevations, especially on the back of the head. Table 2 shows the mean and standard deviation values of the SD for $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$ in different frequency bands. In general, mean SD progressively increases from circa 1.6 dB to more than 6 dB as the frequency increases. Similarly, the standard deviation is larger for higher frequency bands. The mean SD of $\hat{\mathbf{H}}$ is lower for each frequency band with the larger differences found between 8 and 12 kHz, though below 0.5 dB.

[3] Distributions normality hypothesis was rejected by Shapiro-Wilk test. Significant difference between the conditions was found through non-parametric Friedman test. Paired Wilcoxon signed-rank tests were performed to assess the significance of the differences between pairs of conditions. The obtained p-values were adjusted with Holm-Bonferroni correction for the multiple comparisons.

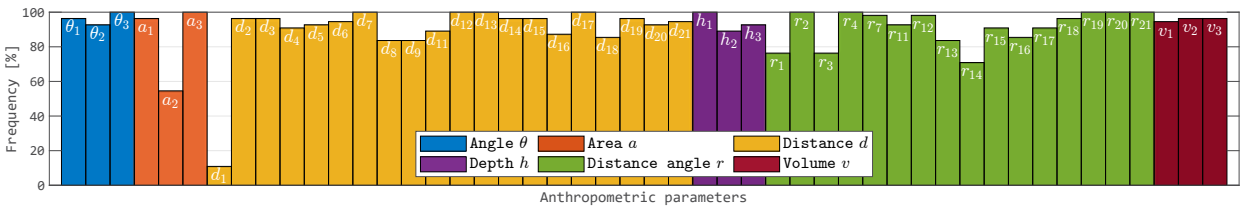


Figure 2. Selection frequency of each parameter obtained in the LOOCV for the condition $\hat{\mathbf{H}}$.

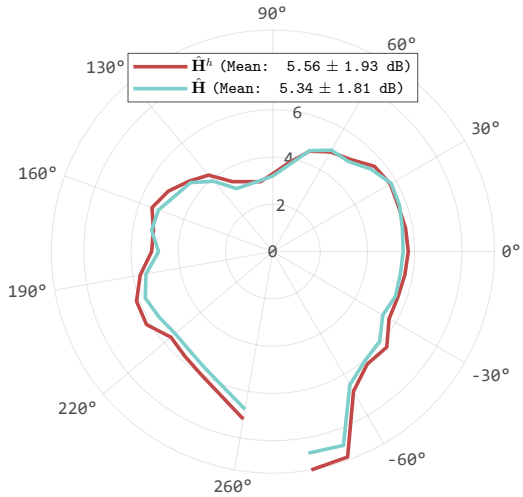


Figure 3. Elevation-dependant mean SD values for the PRTFs $\hat{\mathbf{H}}^h$ selected using HUTUBS anthropometry $\mathbf{A}^h$ (red), and $\hat{\mathbf{H}}$ selected using all parameters $\mathbf{A}$ (light blue).

	Frequency band [kHz]					
	3–4	4–6	6–8	8–10	10–12	12–15
$\hat{\mathbf{H}}^h$	1.64 (1.95)	2.57 (2.36)	4.28 (3.39)	5.53 (3.69)	5.88 (3.47)	6.39 (3.64)
$\hat{\mathbf{H}}$	1.61 (2.09)	2.51 (2.23)	4.09 (3.32)	5.07 (3.51)	5.57 (3.31)	6.21 (3.52)

Table 2. Mean and standard deviation (between brackets) of SD in dB for each frequency band compared for the PRTFs $\hat{\mathbf{H}}^h$ selected using HUTUBS anthropometry $\mathbf{A}^h$ and $\hat{\mathbf{H}}$ selected using all anthropometry $\mathbf{A}$.

5.4 Auditory model evaluation

SD only evaluates the spectral differences between PRTFs and is unsuitable for measuring localization accuracy. Therefore, we compared the selected PRTFs in a virtual localization experiment using the computational auditory model proposed by Baumgartner et al. [37] provided in the auditory modeling toolbox (AMT) [38]. This model estimates the localization performances in the median plane with a template DTF compared to the individual DTF of a subject (target). Several metrics can be computed given actual angles and the subject's responses, and we selected the local polar RMS error (PE) and the quadrant error rate (QE) [39]. PE is the RMS average in degrees of the polar errors with less than 90° in magnitude. QE is the percentage of polar errors higher than 90° in magnitude. We set the frequency range and elevation angles considered by the auditory model according to the ranges defined in the preprocessing step (see Sec. 4.1.2). The sensitivity of the auditory model was set for each subject using the calibration function of Baumgartner's model. We selected the baseline PE and QE values of 25° and 5%, respectively.

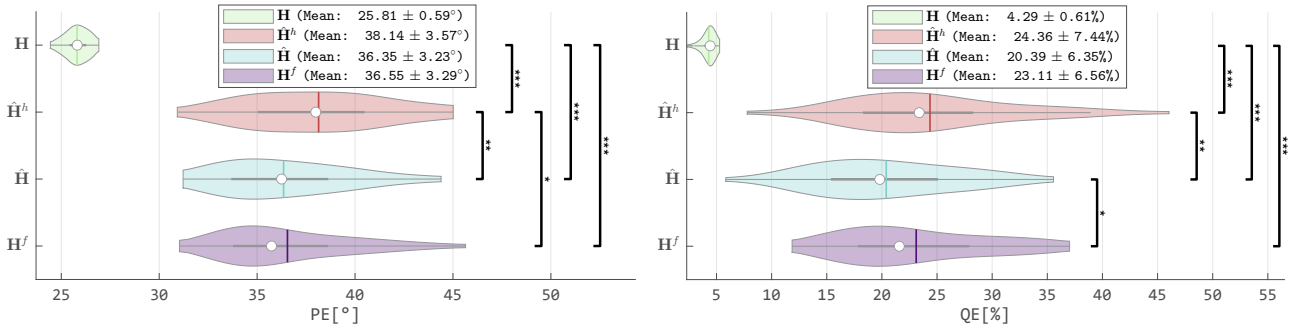
PE and QE values were obtained using the individual PRTF $\mathbf{H}$ as target and other four conditions as templates: the same PRTF $\mathbf{H}$, the selected PRTFs $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$, and the FABIAN PRTF $\mathbf{H}^f$. The distributions of PE and QE for each condition are shown in Fig. 4. A similar statistical analysis to the aforementioned one for SD was conducted[3]. As expected, the individual PRTF $\mathbf{H}$ performs significantly better than any other condition. Then, the mean PE and QE of $\hat{\mathbf{H}}$ are significantly lower than those of $\hat{\mathbf{H}}^h$ ($p < 0.01$) with an improvement of 1.79° and 3.97%, respectively. We found a medium effect size for these differences in PE and QE distributions corresponding to a Cohen's d of 0.52 and 0.57, respectively. We notice that $\mathbf{H}^f$ has performances similar to $\hat{\mathbf{H}}^h$ and $\hat{\mathbf{H}}$. For PE, $\hat{\mathbf{H}}^h$ is significantly higher than $\mathbf{H}^f$ ($p < 0.05$), while $\hat{\mathbf{H}}$ significantly outperforms $\mathbf{H}^f$ only for QE ($p < 0.05$).

6. CONCLUSIONS

In this paper, we proposed an expansion to the pinna anthropometric parameters commonly used in literature for a more comprehensive description of the pinna in the PRTF individualization field. The parameters were measured using landmarks manually annotated on the pinna shape. We compared the proposed parameters to the HUTUBS ones in a PRTF selection task in the median plane. We found a limited, yet statistically significant, improvement in using the proposed parameters for both SD and localization metrics obtained with an auditory model. Given the limited improvement, further investigation is necessary to assess the relevance of the proposed parameters. This work represents an initial step toward an improved pinna characterization, as the baseline approach used for PRTF selection may not be suitable to exploit the information provided by the proposed parameters. Thus, exploring advanced methods for HRTF individualization, such as those based on machine learning, could be beneficial. In future work, we also plan to automate the landmarks placement on pinna images as done in our previous work [26]. This permits to measure the anthropometry without manual intervention.

Acknowledgments

This work is part of SONICOM, a project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017743.



(a) Local polar RMS error (PE)

(b) Quadrant error rate (QE)

Figure 4. Results of the virtual localization experiment through the auditory model [37] in terms of (a) PE and (b) QE distributions. The compared PRTF conditions are: individual $\mathbf{H}$ (green), $\hat{\mathbf{H}}^h$ selected using the HUTUBS anthropometry $\mathbf{A}^h$ (red), $\hat{\mathbf{H}}$ selected using all anthropometry $\mathbf{A}$ (light blue), and FABIAN $\mathbf{H}^f$ (purple). The white dot in the violin plot represents the median, while the vertical colored line is the mean and the horizontal thick gray line is the interquartile range. The vertical black lines between distributions represent a significant difference, while the stars on their right represent the corresponding p-values (*: $p < 0.05$, **: $p < 0.01$, ***: $p < 0.001$).

7. REFERENCES

[1] B. Xie, *Head-Related Transfer Function and Virtual Auditory Display*, 2nd ed. Florida, US: J. Ross Publishing, 2013.

[2] S. Xu, Z. Li, and G. Salvendy, "Individualization of head-related transfer function for three-dimensional virtual auditory display: A review," in *Virtual Reality*, R. Shumaker, Ed. Springer Berlin Heidelberg, 2007, pp. 397–407.

[3] E. M. Wenzel, M. Arruda, D. J. Kistler, and F. L. Wightman, "Localization using nonindividualized head-related transfer functions," *The Journal of the Acoustical Society of America*, vol. 94, no. 1, pp. 111–123, 1993.

[4] H. Møller, M. F. Sørensen, C. B. Jensen, and D. Hammershøi, "Binaural technique: Do we need individual recordings?" *Journal of the Audio Engineering Society*, vol. 44, no. 6, pp. 451–469, 1996.

[5] D. R. Begault, E. M. Wenzel, and M. R. Anderson, "Direct comparison of the impact of head tracking, reverberation, and individualized head-related transfer functions on the spatial perception of a virtual speech source," *Journal of the Audio Engineering Society*, vol. 49, no. 10, pp. 904–916, october 2001.

[6] R. Bomhardt, "Anthropometric individualization of head-related transfer functions analysis and modeling," Ph.D. dissertation, Institute of Technical Acoustics, Aachen University, Berlin, Germany, 2017.

[7] C. Guezenoc and R. Seguier, "HRTF individualization: A survey," in *Proc. AES Convention*, New York, 2018.

[8] V. R. Algazi, R. O. Duda, D. M. Thompson, and C. Avendano, "The CIPIC HRTF database," in *Proceedings of the 2001 IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (Cat. No. 01TH8575)*. IEEE, 2001, pp. 99–102.

[9] P. Stitt and B. F. Katz, "Sensitivity analysis of pinna morphology on head-related transfer functions simulated via a parametric pinna model," *The Journal of the Acoustical Society of America*, vol. 149, no. 4, pp. 2559–2572, 2021.

[10] F. Brinkmann, M. Dinakaran, R. Pelzer, P. Grosche, D. Voss, and S. Weinzierl, "A cross-evaluated database of measured and simulated HRTFs including 3D head meshes, anthropometric features, and headphone impulse responses," *Journal of the Audio Engineering Society*, vol. 67, no. 9, pp. 705–718, 2019.

[11] H. Takemoto, P. Mokhtari, H. Kato, R. Nishimura, and K. Iida, "Mechanism for generating peaks and notches of head-related transfer functions in the median plane," *The Journal of the Acoustical Society of America*, vol. 132, no. 6, pp. 3832–3841, 2012.

[12] M. Burkhard and R. Sachs, "Anthropometric manikin for acoustic research," *The Journal of the Acoustical Society of America*, vol. 58, no. 1, pp. 214–222, 07 1975.

[13] J. C. Middlebrooks, "Individual differences in external-ear transfer functions reduced by scaling in frequency," *The Journal of the Acoustical Society of America*, vol. 106, no. 3, pp. 1480–1492, 09 1999.

[14] S. Ghorbal, X. Bonjour, and R. Séguier, "Computed HRIRs and ears database for acoustic research," in *Audio Engineering Society Convention 148*. Audio Engineering Society, 2020.

[15] N. S. A. o. C. GB/T 2428-1998, "GB/T 2428-1998 standard head-face dimensions of adults," General administration of quality supervision and inspection of China, Tech. Rep., 1998.

[16] Z. Guo, Y. Lu, H. Zhou, Z. Li, Y. Fan, and G. Yu, "Anthropometric-based clustering of pinnae and its application in personalizing hrtfs," *International Journal of Industrial Ergonomics*, vol. 81, p. 103076, 2021.

[17] B. Xie, X. Zhong, D. Rao, and Z. Liang, "Head-related transfer function database and its analyses," *Science in China Series G: Physics, Mechanics and Astronomy*, vol. 50, no. 3, pp. 267–280, 2007.

[18] K. Iida, O. Nishiyama, and T. Aizaki, "Estimation of the category of notch frequency bins of the individual head-related transfer functions using the anthropometry of the listener's pinnae," *Applied Acoustics*, vol. 177, p. 107929, 2021.

[19] Y. Teng and X. Zhong, "An individualized HRTF model based on random forest and anthropometric parameters," in *2023 15th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*. IEEE, 2023, pp. 143–146.

[20] E. A. Shaw and R. Teranishi, "Sound pressure generated in an external-ear replica and real human ears by a nearby point source," *The Journal of the Acoustical Society of America*, vol. 44, no. 1, pp. 240–249, 1968.

[21] V. R. Algazi, C. Avendano, and R. O. Duda, "Elevation localization and head-related transfer function analysis at low frequencies," *The Journal of the Acoustical Society of America*, vol. 109, no. 3, pp. 1110–1122, 2001.

[22] V. C. Raykar, R. Duraiswami, and B. Yegnanarayana, "Extracting the frequencies of the pinna spectral notches in measured head related impulse responses," *The Journal of the Acoustical Society of America*, vol. 118, no. 1, pp. 364–374, 2005.

[23] M. B. Gardner and R. S. Gardner, "Problem of localization in the median plane: effect of pinnae cavity occlusion," *The Journal of the Acoustical Society of America*, vol. 53, no. 2, pp. 400–408, 1973.

[24] J. Fels and M. Vorländer, "Anthropometric parameters influencing head-related transfer functions," *Acta Acustica united with Acustica*, vol. 95, no. 2, pp. 331–342, 2009.

[25] S. Ghorbal, T. Auclair, C. Soladie, and R. Seguier, "Pinna morphological parameters influencing HRTF sets," in *Proceedings of the 20th International Conference on Digital Audio Effects (DAFx-17)*, 2017.

[26] D. Fantini, F. Avanzini, S. Ntalampiras, and G. Presti, "HRTF individualization based on anthropometric measurements extracted from 3D head meshes," in *2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*. IEEE, 2021, pp. 1–10.

[27] M. Dinakaran, P. Grosche, F. Brinkmann, and S. Weinzierl, "Extraction of anthropometric measures from 3D-meshes for the individualization of head-related transfer functions," in *Audio Engineering Society Convention 140*. Audio Engineering Society, 2016.

[28] E. A. Torres-Gallegos, F. Orduna-Bustamante, and F. Arámbula-Cosío, "Personalization of head-related transfer functions (HRTF) based on automatic photo-anthropometry and inference from a database," *Applied Acoustics*, vol. 97, pp. 84–95, 2015.

[29] P. Majdak, M. J. Goupell, and B. Laback, "3-D localization of virtual sound sources: Effects of visual environment, pointing method, and training," *Attention, Perception, & Psychophysics*, vol. 72, no. 2, pp. 454–469, 2010.

[30] K. Iida, H. Shimazaki, and M. Oota, "Generation of the amplitude spectra of the individual head-related transfer functions in the upper median plane based on the anthropometry of the listener's pinnae," *Applied Acoustics*, vol. 155, pp. 280–285, 2019.

[31] X. Liu, W. Huang, H. Zhang, and X. Zhong, "Median-plane head-related transfer function personalization using two-dimensional independent component analysis," in *2022 IEEE 8th International Conference on Computer and Communications (ICCC)*. IEEE, 2022, pp. 2308–2312.

[32] M. G. Onofrei, R. Miccini, R. Unnthorsson, S. Serafin, and S. Spagnol, "3D ear shape as an estimator of HRTF notch frequency," in *Proceedings of the 17th Sound and Music Computing Conference (SMC 2020)*. Axea sas/SMC Network, Jun 2020, pp. 131–137.

[33] D. Zotkin, J. Hwang, R. Duraiswaini, and L. S. Davis, "HRTF personalization using anthropometric measurements," in *2003 IEEE workshop on applications of signal processing to audio and acoustics (IEEE Cat. No. 03TH8684)*. IEEE, 2003, pp. 157–160.

[34] M. Geronazzo, S. Spagnol, and F. Avanzini, "Mixed structural modeling of head-related transfer functions for customized binaural audio delivery," in *2013 18th International Conference on Digital Signal Processing (DSP)*. IEEE, 2013, pp. 1–8.

[35] B. Zhi, D. N. Zotkin, and R. Duraiswami, "Towards fast and convenient end-to-end HRTF personalization," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 441–445.

[36] J. Cohen, *Statistical Power Analysis for the Behavioral Sciences*. England, UK: Routledge, 1988.

[37] R. Baumgartner, P. Majdak, and B. Laback, "Modeling sound-source localization in sagittal planes for human listeners," *The Journal of the Acoustical Society of America*, vol. 136, no. 2, pp. 791–802, 2014.

[38] Majdak, Piotr, Hollomey, Clara, and Baumgartner, Robert, "AMT 1.x: A toolbox for reproducible research in auditory modeling," *Acta Acustica*, vol. 6, p. 19, 2022.

[39] J. C. Middlebrooks, "Virtual localization improved by scaling nonindividualized external-ear transfer functions in frequency," *The Journal of the Acoustical Society of America*, vol. 106, no. 3, pp. 1493–1510, 1999.

A SPEAKER AGNOSTIC APPROACH TO SPATIALISATION IN ELECTROACOUSTIC MUSIC

Stefano CATENA (stefano.catena23@gmail.com) (0000-0001-7650-4203)[1] and
Henrik FRISK (henrik.frisk@kmh.se) (0000-0003-1958-8484)[2]

[1]**De Montfort University**, *MTI*[2] *- Music, Technology and Innovation - Institute for Sonic Creativity*, Leicester, United Kingdom
[2]**Royal College of Music in Stockholm (KMH)**, Stockholm, Sweden

ABSTRACT

While spatial audio technologies have developed dramatically in the last twenty years, the compositional approaches to spatialisation in electroacoustic music are still fragmented. No coherent and shared language for spatialisation exists yet, and it is often challenging to reproduce an electroacoustic piece in different space from the one it was conceived in. In this paper we delve deeper into an approach about spatial audio and spatialisation techniques in the electroacoustic arts that we called 'speaker agnosticism'. By introducing Lilla Salen, a multichannel system of up to 49 speakers in a dome like configuration, and the authors' compositions, we describe a method of treating the concert hall as an instrument irrespective of the number of speakers it contains, or their arrangement. This paradigm shift encourages an 'organic' approach to music spatialisation, steering away from the traditional 'Euclidean' mindset. Finally, this project hints at the possibility of 'touring' pieces: by treating multichannel systems rather as instruments. Thanks to current audio technologies it is possible to 'transcribe' works between concert halls without losing the musical intention.

1. INTRODUCTION

It is almost impossible to divorce space from sound. Even a sound emanating from an anechoic chamber communicates a lack of space so remarkable that the lack of space itself becomes its identity. When listening to a piano in a concert hall, or when we hear a bird singing in the forest, whatever spatial information is in the sound - the reverberation of the concert hall or the density of the forest - is embedded and becomes an integral part of the perception of the sound. Yet, in musicology the theoretical and analytical representations of musical elements has for centuries often been reduced to quantitative parameters such as pitch (frequency) and rhythm (time). Still, from a perceptual point of view, the space of a sound is a parameter intimately tied to its cognitive meaning, to the degree that two sounds that share the same rhythm, pitch and timbre may be interpreted very differently by a listener if they are spatially different. Many of the technologies developed for working with spatial audio are biased towards a particular understanding of what spatial audio is. This is even more noticeable in electroacoustic music, where the spatial component of sound is one of its key elements [1].

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. BACKGROUND

Since the introduction of the loudspeaker, spatialisation and the spatiality of sound have taken an increasingly important role in the electroacoustic arts. Works such as *Gesang der Junglinge* (1955-56) or *Kontakte* (1958-60) from Karlheinz Stockhausen make use of the technology present in the 1950's to create an immersive musical scene and to exploit spatiality for a musical goal. *Gesang der Junglinge* uses five groups of spatially separated loudspeakers [2], while *Kontakte* uses a four-channel speaker layout to project sound. Another notable example is Xenakis' *Concrète PH* (1958) and the Bruxelles Pavillion for which he contributed a sound installation, where over 400 loudspeakers were positioned inside the building in order to create spatial trajectories and movements [3]. In parallel, the acousmatic French school developed instruments and tools for the diffusion of electroacoustic composition in music halls: Pierre Henry and Jaques Poullin introduced in 1951 the potentiometre d'espace to manage sound diffusion, but it is only in the 1970's that Francois Bayle uses an 'orchestra of loudspeakers', the Acousmonium to 'interpret' sound spatialisation in a concert setting [4]. John Chowning's experiment with computer music and Doppler effects are also a notable example of historical use of spatialisation: his piece *Turenas* (1971), for instance, is a four-channel composition that actively includes spatial cues to 'trick' the listener into perceiving the direction of a simulated source [5]. Also in the 70's was the first-order Ambisonics was conceived by Gerzon where a B-Format audio signal is used to represent the sound field instead of the speaker information, and later decode such signals for any arbitrary speaker layout [6]. Today this a very common technology for spatialisation, recommended by internet content providers such as Google or YouTube. [1] First Order and Higher Order Ambisonics (FOA and HOA) are some of the most popular spatialisation technologies used

[1] https://resonance-audio.github.io/resonance-audio/discover/concepts

currently [7], even though in the last years technologies like Dolby Atmos have caught up remarkably in the electroacoustic arts, too.

There is no shortage of research on the technologies of spatialisation and even though the situation has improved slightly, Blesser and Salter's statement that "although there is a vast body of scholarly work both on the physical acoustics of enclosed spaces and on perceiving acoustic parameters, the literature is relatively silent on the subject of how people experience aural space" still has some validity. They continue: "We know much about measuring acoustic processes and sensory detection, but less about the phenomenology of aural space" [8]. Malham and Myatt comments that historically, the greatest problem with spatialised music has been the lack of, not only a comprehensive theory of spatialisation, but also of simple means to control it. Although much work has since been done, there are still no obvious standards [9]. Ambisonics, a method for sound spatialisation, is grounded in a solid theory, but a comprehensive understanding of sound localization should include a phenomenological knowledge of the relation between sound and space [9–11]. Hence, if the early experiments beginning in the 50's lacked standardized control systems and theory, much of which has been settled in the last two decades, a broad and discursive aesthetic theory based on practical, artistic applications of sound diffusion is still lacking. Nowadays, the affordability of computing power and loudspeaker setups has increased the interest for large multichannel reproduction systems: one of the most famous is the BEAST (Birmingham ElectroAcoustic Sound Theater) that has evolved from its original 'main eight' to over 100 speakers [12]. Another notable example is Lilla Salen at the Royal College of Music in Stockholm (KMH) or the Sonosfera in Pesaro with 45 custom built loudspeakers [2].

While the technological advancements in spatial audio technologies have been remarkable and permitted a larger pool of composers and artists to access complex spatialisation techniques, no coherent and shared musicological framework has been developed to describe their musical use [13]. However, several theories and ontologies of space have been proposed: one of the most prominent is Smalley's 'spatiomorphology' [1] and his investigation on musical spaces [14]. His concepts of 'composed space', defined as "the space composed on the fixed media", and 'listening space', where the composition is played back into, are particularly relevant. In our investigation, only the concept of 'composed space' is taken into consideration: our range of action is limited to the space as created and 'recorded' by the composer, and all other approaches to spatialisation (e.g. live diffusion, multimedia installations etc) will not be taken into consideration. In his 'On Sonic Art', Wishart proposes and investigates a series of spatial motions, their combinations in a sort of 'spatial counterpoint' and their perceptual relevance [15]; Kendall tries to describe and categorize the attributes of 'spatial imagery' [16] and Catena attempts to establish an atomic unit for the description of spatialisation [17]. It is also important to clarify terminology while talking about spatialisation: many times terms like 'spatial audio' and 'spatiality' may be confusing or wrongly used. Holbrook describes the term 'spatial audio' as a "set of tools and methods for how to represent and control sound material through a process of spatialisation" [18]. Consequently, we can define spatialisation as the conscious compositional act of moving or placing sounds inside or outside a given space with an intended musical aim. While this definition is similar to Holbrook's, he omits to recognize that instrumental music can also employ spatialisation techniques to convey a musical meaning. The term 'spatiality', instead, is used to describe a set of physical spatial qualities of sound, e.g. its reverberation, perceived distance etc..

Another problem of spatialisation and spatial music in the electroacoustic arts is its reproducibility: once a piece is composed in a studio or in a particular venue, it has been historically complicated 'moving' it from one place to the other. With amplitude based spatialisation technologies the number of channels in the audio file represents the number of physical speakers they need to address: a piece written for a 9.1 setup could not be faithfully reproduced in an octophonic ring. By adopting the concept of speaker agnosticism we intend to address a compositional challenge as well as attempt to develop a preliminary terminology for the field of spatialisation studies, i.e. the way spatial properties in music can be understood and conceptualized, by mainly focusing on a set of attributes of perceptual importance. The term 'agnostic' has been preivously used to describe the MPEG-H standard for audio encoding and object-based spatialisation [19], but no study exists that has gone into details of what speaker agnosticism entails artistically. By means of focusing on spatial perception rather than using a Euclidian [3] approach to spatialisation we aim to suggest a theory of speaker agnostic composition practice.

3. SPEAKER AGNOSTICISM

The spatial musical aspect of a piece does not necessarily need to take into consideration the actual physical speaker layout in the room in which it is going to be played in. Ambisonics is a technology that is speaker agnostic at its core since it represents a sound field excitation as spherical harmonics [6], rather than hard coding spatial information into the different channels. Regardless of the technology, which may vary in the future, the most fascinating aspect is how speaker agnosticism influences the process of composition. As is found by Schumacher: "it is generally difficult to recognize sound trajectories in space when there are no supporting visual stimuli" [20], which makes Euclidian approaches for sound spatialisation musically ineffective. Yet, a Euclidean approach to space is most commonly used as the basis for the spatialisation interface both for Ambisonics and many other spatialisation technologies.

Speaker agnosticism is the notion that assumes that spa-

[2] sonosfera.eu

[3] By 'Euclidian' we intend a spatialisation method that relies on a geometrical set of values to describe the instantaneous positions of sound objects in space. This concept will be described in more detail in section 3.1

tiomorphologies [1] are independent of reproduction and encoding technologies, inducing a 'relativist' spatial thinking [21]. For example, it is more important that a sounds moves, rather than defining a precise spatial trajectory that is difficult to locate and perceive, perhaps distorting the real compositional intent. This relates to the concrete perception of the morphology of its placements or movements, and it is only relevant in conjunction with consideration of the sound and the feeling that the sonic material may evoke. For example, a sound coming from behind or from the sides may be perceived as scary and unexpected (a howling wolf from behind the listener can be described as eerie), while a sound placed in the front may evoke a sense of anticipation, engagement or clarity. However, going even further with a speaker agnostic mindset, one could even disregard the traditional concept of front and back or left and right. If all directions are deemed equally significant, only the listener's subjective location becomes crucial, making the listener experience the primary focus of the creative process.

3.1 Organic vs Euclidean

To describe the difference between Euclidean and organic thinking of spatialisation it is useful to start with a practical example. Let's imagine to stand in a forest and to think about what we could hear: an open space, the rustling of foliage, birds singing and flying, cracking branches, insects buzzing and the sound of flowing water of a nearby river. In an organic way of thinking spatialisation of such sound scene one could differentiate between three main categories of spatiomorphologies:

- *Movements*: sound motions, be they continuous or discontinuous, where a clear, perceivable, and coherent change in location of the sonic material is heard. Sound trajectories are a common example of spatial movements: the sound of flapping wings of a flying bird, or a the buzzing of a passing wasp. A different example, in a more musical context, can be seen in Barry Truax's 'Shaman Ascending' [4]: the whole piece is based on a rotatory movement of a vocal sound, until the rotation becomes so fast that it cannot be heard anymore. Being composed for an 8-channel system, the sound is moved intermittently between speakers, rather than blended between them to create a smooth transition. From a perceptual point of view, I regard these two approaches similar: both create the perception of a motion, be it continuous or discontinuous, and therefore falling in the same category.

- *Placements*: absence of motion. Whenever a sound is statically located and its position easily perceivable, we would call this spatiomorphology a placement. For example, the identifiable high frequency sounds of birds songs where they can be heard in their immovable position.

[4] https://www.sfu.ca/ truax/shaman.html

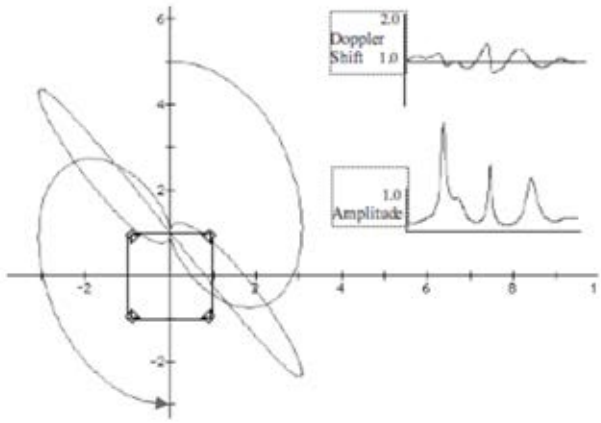


Figure 1. One of the Lissajous figure in *Turenas*, with Doppler effect and Amplitude values [22]

- *Occupancy*: a diffused positioning of sound. For example the immersiveness of wind passing through foliage creating the perception of sound coming from all around the listener and effectively occupying a portion of (or the whole) spatial scene.

When thinking about sound in a 'spatial' manner, these are the categories in which spatiomorphologies can be formed concretely. Their combination creates the complex spatial structure of a piece, and their function and relation to the sound material generates the musical discourse. Moreover, it is not always important to think precisely *how* these spatiomorphologies behave: for example, it is musically more relevant that a sound moves, rather than it moving from left to right (or viceversa); or that this sounds is placed in a specific location, rather than on its opposite side (even if this may not always be the case). However, in an organic thinking of spatialisation, it is the movement, the placement or the occupancy of sound that is spatially meaningful, and how the combination of spatio and spectromorphologies results in the creation of the musical structure.

Conversely, a Euclidian approach to space considers spatiomorphologies of a sound source described only by a set of geometrical coordinates, rather than looking at their spatial qualities. For example, in the iconic John Chowning's piece *Turenas*, the composer creates spatial illusion of moving sources, along with Doppler effect, azimuth and distance manipulation [22]. Through his system, Chowning was able to design complex trajectories of sounds in space, following Lissajous figures (as shown in Figure 1) in a very precise geometric space. However, this approach was used in a quadraphonic setup, known for being an unreliable reproduction system for sound localisation and, more importantly, it would be impossible to accurately recognize and track the Lissajous figures, making them musically ineffective.

These two methods for the compositional thinking of spatial music can be interpreted in terms of 'relativist' and 'absolutist'. The organic approach can be described as more 'relativist', where macro categories of spatiomorphologies are used to musically exploit their perceived spa-

tial shapes and characters; the traditional Euclidian approach can be view as 'absolutist', where the spatial placements and movements follow a precise and 'objective' set of numeral descriptors, creating complex spatial structures. Our intention is to return to the subjective - i.e. organic - listening experience as the key source of knowledge for sound spatialisation in electroacoustic composition. This also includes the rejection of the power of the concert hall and technologically advanced multi speaker environments as an idealized place for virtual sound worlds, and instead make the act of composition the main spatial listening exercise. In this re-framed performance environment various forms of listening hold a central aspect of the spatiality of the music. Our concept of speaker agnosticism tries to shift the focus from the technology to the technique and to the listening experience. However, not all organic or Euclidian approaches to spatialisation are necessarily speaker agnostic: these two ways of thinking the space relate only to how the composer approaches and creates the 'composed space'. In fact, speaker agnosticism relies on a suitable technology that permits a 'translation' between different multichannel systems, such as Ambisonics or Dolby Atmos.

3.2 From technique to technology

Most of the technology (both hardware and software) for spatialisation today follow a Euclidian approach to space. Whether it be Ambisonics, DBAP (Distance Based Amplitude Panner), VBAP (Vector Based Amplitude Panning), Dolby Atmos or others, one would have to specify geometrical coordinates for the positioning of the sound sources in space. This affects the way in which a composer places and moves sound in the piece. Moreover, due to this technological factor, it is inevitable to deal with some form of geometrical aspects during the spatisalisation act. One cannot avoid to specify azimuth angles or cartesian coordinates in their compositional workflow. However, if one were to establish and to think of general directions instead of geometrical coordinates, a different perspective for spatialisation arises. The perspective of a continuous space, rather than a discrete one, where the compositional process follows meaningful perceptual properties. This compositional process, in fact, would help and contribute to a speaker agnostic approach to spatialisation, since no technical information is required. By disregarding the intention of excessively localised trajectories and sound placements, it is possible to think of an idealised empty space to fill, rather than an actual physical multichannel system with speakers to address. Furthermore, by detaching the technological aspects from the work (i.e. thinking agnostically), the importance of the musical act is given back to the spatiomorphologies - and consequently to the listening experience. By using an algorithmic approach, for example, it would be easy to displace sound into perceptually relevant 'macro-directions': a relevant example is described in section 4.1.1.

4. COMPOSITIONAL PRACTICE

4.1 Travelling without moving

Travelling without moving is an acousmatic composition written at the KMH in Stockholm in 2023, specifically in the multichannel half dome present in Lilla Salen, that strongly incorporates the 'composed space' as one of its most important features. As the title suggests, the piece revolves around the concept of motion, movement or their absence: most of the sonic material can be heard while traversing the performance space, creating complex spatial textures and perspectives. However, in several key moments, the strategic static placement of the material will clash, contrast or complement the rest of the spatial scene. Moreover, another key component of the piece is the relationship between timbre and space: 'how the sound behaves' and 'how the sound moves' are the two intertwined ideas that are investigated musically. All the employed timbres come from the Buchla modular system (Figure 2) present in the electroacoustic facilities of KMH. (insert image). Modulated oscillators (Dual Oscillator 258 and Quad Function Generator 281e) and noise generators (Source of Uncertainty 266) in conjunction with the Buchla's 'Spectral Processor 296e' were utilized to produce and shape the timbral material of the composition.

Figure 2. The Buchla system present at the Royal College of Music facilities

Algorithmic manipulation of the sonic matter and tape techniques were employed to further modify and recombine the recorded timbres as needed. The spectromorphologies that resulted from the sound generation and its consequent transformation have deeply influenced the compositional choices regarding spatialisation: the sonic material and its deployment in space are strongly linked and not independent from one another. For example, pitch

and amplitude changes are most likely linked with a moving source that follows along the spectromorphological changes, while sounds that are strongly connected to natural environments (e.g. rain) would occupy large portions of the spatial scene, enveloping the listeners. The composition is structured in three main sections that follow one another in sequence:

- *Departure*: the first section consists mostly of rotary timbres, similar to how a helicopter whirring would sound. These sound sources are split into several bands, where each portion of the spectrum is moved around the listener to create an enveloping texture of moving objects. This first section represents a departure from the real-world and into another sonic and spatial environment, where the mind is free to travel where it wants.
- *Travelling*: when the first section ends, relaxing sounds of falling rain fade in and fully occupy the scene, just like waking from a dream after sleeping too much on a green grass field. From this soundscape various wandering objects emerge, collide and, through a process of accumulation and layering, create a complex texture of moving sounds from and to all directions, similar to a storm. After this storming climax, a slow and gradual decrescendo is present, leading to the next section. Both the climax and the decrescendo involve the transformation of all musical parameters: timbre, space, dynamic and pitch.
- *Landing*: the last section calls back to the relaxing sound of falling rain, but with the addition of several statically positioned sounds: insects and birds chirping are added to the soundscape, finally waking up the listener. Moving textures recall the travel and its climax, but they fade away like a forgotten dream. Eventually, within the synthesized sound of rain, a paradoxical 'immobile' sound of footsteps is heard, gradually dying out and closing the piece.

Ideally, this piece should be played in venues that can accommodate multichannel reproduction in an hesa/octophonic setup at minimum; however, the piece is not strictly tied to a particular space or speaker. Thanks to Ambisonics technology, it is possible to decode the piece in any speaker layout present at any time, making the piece, by definition, speaker agnostic. The composition can be heard online in a binaural rendition. [5]

4.1.1 Approach to spatialisation in Lilla Salen

As mentioned, *Travelling without moving* has been composed at the Royal College of Music in Stockholm, more precisely with the multichannel half dome system in the concert hall called Lilla Salen.

Lilla Salen at KMH (shown in Figure 3) is configured for both research and experimental performance. A flexible system of up to 49 loudspeakers in a dome like configuration affords a platform for a wide variety of projects

Figure 3. Lilla Salen

including sound spatialisation either in the acousmatic tradition or with acoustic and electronic instruments together. The system creates an immersive sound environment while providing precise sonic imaging for studies in spatialisation and cognition. Merely describing a space like Lilla Salen reveals the focus on the technological aspect of the room and for the composer working in this space it tends to put the focus on how to handle the speaker layout and the structurality of the technology in the hall. How the various inputs and outputs are configured tends to allow, or disallow, particular practices. The vision that a spatialisation technique such as Ambisonics fosters that spatial music can be agnostic towards speaker layouts and performance spaces is challenged by this. To instead treat the hall as an instrument, a specificity rather than a general space for listening and creation, is an alternative approach. Then any new rendering of such a piece for a different hall is an interpretation much more than an exact replica. In fact, the spatial approach and thinking while composing the piece does not take into consideration the technical aspects of the hall, but rather exploits and treats the space for what it is: a half dome.

A speaker agnostic thinking of the space encourages an algorithmic approach to spatialisation. With the term 'algorithmic' we intend "the application of rigid, well-defined algorithms to the process of composing music" [23]. By generating several takes of spatialised material, it is crucial to 'compose' the desired texture after a critical listening. This algorithmic method allows to not think of space mathematically - i.e. Euclidian thinking - but rather to listen to how the space is generated, how the sounds moves, how it occupies portions of the space, the interconnections created between spectro and spatiomorphologies. Ambisonics is the ideal technology for this approach, as it is able to intertwine sound and space at their inception and decode the spatial information for any kind of speaker layout. Moreover, its flexibility and ease of use has made it the optimal companion of programming langugages such as Supercollider [6] or Max/MSP from Cycling 74 [7] . In this case, Supercollider was the preferred language from the author, since it has historically handled multichannel signals reliably.

This method has been used all throughout *Travelling without moving*. From 2:15 to 5:20, for instance, all the spatialisation adopts exclusively this approach, creating

[5] https://soundcloud.com/stefanocatena/travelling-without-moving-binaural-version

[6] https://supercollider.github.io

[7] https://cycling74.com

```
1  ( //front TO back, down to up -- general directions
2  {
3      256.do({ //256 objects
4          {
5              HOAEncoder.ar(3, //High Order Encoder, 3rd
6                  LPF.ar( //Lowpass filter
7                      PlayBuf.ar( //Playing audio from Buffer
8                          1, //mono buffer
9                          b, //actual buffer
10                         rrand(0.1, 0.3) + LFNoise2.kr(5).range(0.01, 0.1),
11                         //rate with randomness control
12                         1, //trigger
13                         rrand(SampleRate.ir * 70, SampleRate.ir * 80)
14                         //starting position for buffer reproduction
15                     )  *
16                     EnvGen.kr( //Envelope generator for each object
17                         // for amplitude shaping
18                         Env.new(
19                             [0, 1, 0],
20                             [0.75, 0.75],
21                             \lin ), //linear trajectory
22                         1 //gate of the envelope - always on
23                     ) * -6.dbamp, //Needed amplitude scaling
24                     1000 //Cutoff Frequency
25                 ),
26                 Line.kr( //spatial positioning - Lines
27                     (rrand(0, 10) * [-1, 1].choose).degrad,
28                     //generating from the front, approximately
29                     (rrand(170, 180) * [-1, 1].choose).degrad,
30                     //ending to the back, approximately
31                     1.5, //duration of the movement
32                     doneAction:2 //killing the synth
33                 ),
34                 Line.kr(0, pi/4, 1)) //altitude
35             //moving from down to up in 1 second
36         }.play; //play it!
37         rrand(0.1, 0.2).wait; //random waiting between objects
38     });
39 }.fork;
40 )
```

Figure 4. Example of a routine that generates complex spatial textures from single moving objects using Higher Order Ambisonics in Supercollider

the ever-moving spatial fabric with several recordings algorithmically generated, carefully chosen, organised and timbrically modified.

From the example shown in Figure 4 it is possible to see how there is no Ambisonics decoder present. This is a very specific compositional choice that leads to an organic process in creating the space. To include a decoder would mean to translate a speaker agnostic signal (the B-format) to a physical speaker layout. By doing this one could be biased to 'compose for the space', rather than 'compose the space': in fact, it could be argued that with this approach we are composing (for) the Ambisonics B-format. In this example, while we inevitably specify geometrical coordinates in the code, it is also noticeable the 'relativism' in creating the spatiomorphologies. The randomness present at lines 27 and 29 from Figure 4 represent the intention of not producing pre-determined specific trajectories, but rather that pseudo-random motions moving from and to macro-directions will form and coalesce into a spatial texture of musical meaning. It is also important to notice that Higher Order Ambisonics has been used: in line 5 of Figure 4 the HOA encoder and its order is specificed. By increasing the Ambisonics order more spherical harmonics are introduced, which increseas spatial accuracy, directional resolution and enlarges the sweet-spot [6]. However, for the Higher Order Ambisonics B-format a large number of channels is required, which equals to $(n+1)^2$ where n is the Ambisonics Order: for instance, a third order B-Format

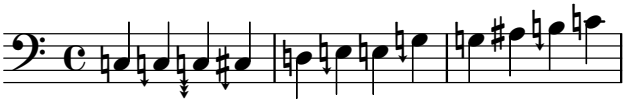


Figure 5. Basic scale for the melodic, harmonic, and spatial structure of *For Bill, Rising*

uncompressed .wav file is made up of 16 channels.

Eventually, the physical attributes of the space will have to be determined: however, this last process can be considered outside of the compositional system, as it is not involved in any way in the writing of the musical piece, but is rather only an infrastructure that aids the listening experience and audio monitoring. In fact, thanks to IEM [8] audio plugins for Ambisonics workflow, one could quickly switch between multichannel or binaural listening, based on contingency or preference. In the case of *Travelling without moving*, the audio monitoring has been primarily provided by the system in Lilla Salen, but in many cases the binaural renderer has helped the process as an immediate and accessible reference. The Ambisonics decoder used for the Lilla Salen monitoring system can be found in the author's Github repository: [9] in this case the IEM AllRad Decoder has been used. [10]

4.2 For Bill, Rising

For Bill, Rising is in a sense inspired by James Tenney's *For Ann, Rising* although it does not have much sonic references to it. The basic idea behind this composition is to use a simple synthesis model, in this case a snare drum, and create timbral variations through combining repetitions of the original sound and spatialising them differently. The tonality and the rhythm is derived from the 7-limit, 11-note scale shown in Figure 5. In the piece rhythms are treated as slowed down intervals and vice versa. The form is structured around a rather simple presentation of the scale and the harmonic and melodic material emanates from it. Originally, the piece is constructed using a variety of different techniques for spatialisation, departing from the notion that each sound has its own spatial properties already at the time of its generation. Hence, the piece is not synthesized and then spatialised, but space is composed into the sound. The technique used for the spatialisation is vector based, but the final representation is rendered in ambisonics. The composition can be heard online in a binaural rendition. [11]

Basically, the piece is a step wise motion from the root of the scale and upwards until the octave is reached at which case the piece has reached its end (See Figure 5). The basic spatial direction, i.e. the centre of all spatial movement, for each of the twelve sections of the piece corresponds to the relative angle of each of the notes in the scale compared to the root. For example, the distance between the root and the first note in the scale is the distance betweeen $\frac{1}{1}$ and $\frac{81}{80}$ (roughly 21.5 cents) which is translated to the

[8] https://plugins.iem.at
[9] https://github.com/friskgit/kmh_ls
[10] https://plugins.iem.at/docs/allradecoder/
[11] https://soundcloud.com/mail-97/for-bill-rising

angle $6.45° = 21.5/1200 * 360°$. In this manner all angles used for spatialisation in the piece are related to the intervals in the scale. In this sense the spatialisation of the piece is dervied from a Euclidean way of thinking. At the same time, however, the spatiomorphology of the piece follows an organic thinking about spatialisation since all *movements* are linked to the *placements* and *occupancies* in the piece. Even though the musical and spatial parameters are mathematically generated the perceptual and organic thinking can be observed here. The focus on the space is composed *through* the material, rather than the material being composed *into* the space. *For Bill, Rising* is to a high degree composed algorithmically in a similar fashion to *Travelling without moving*, as described above. It is an attempt to try to break with the common way of dividing the practice of composition into several interrelated stages and instead approach the synthesis of the material as an integrated practice. In this case it means that the parameters and the data for each musical gesture are used and allowed to influence several musical aspects–e.g. timbre, space, rhythm, harmony, and others–in the same process. This integrated method of composition further emphasizes the organic modality of spatialisation, allowing space to evolve from within the sound, rather than something that is applied on top of the sound. The timbre as well as the spatialisation of the sound are then functions of a mode of composition that departs from algorithmic thinking used to explore the musical material. This participates in integrating spatialisation as a musical parameter along with the others, rather than an aspect that is added at the end. In some sections of the piece spatialisation is used as a synthesis method, further blurring the lines commonly dividing the traditional compositional practice's sub tasks: synthesis, composition, mixing, spatialisation, etc. Just as for *Travelling without moving*, in *For Bill, Rising* this algorithmic process has been achieved using SuperCollider. Although the piece was originally composed for the Lilla Salen hall at KMH, it has been rendered speaker agnostic through Ambisonics. This has been achieved by re-recording it as a B-format Ambisonics. The original piece was rendered using VBAP to a file with 29 channels, one for each speaker in the Lilla Salen Dome. This file was transformed to a B-format signal by feeding each signal through a separate encoder panned to the appropriate position of the speaker in question. The difference between the original version using vector based techniques for spatialisation and the Ambisonics version is noticeable when the two versions are compared in Lilla Salen. Some of the precision of the VBAP panning of single sound sources is blurred by rendering the piece in Ambisonics, as may perhaps be expected given the nature of both the process of encoding and the Ambisonics technology. However, the quality of the spatial impression of the more swirling sounds in the piece, as well as the distant sounds are generally improved in the Ambisonics version. The piece has been performed in a few other locations (Helsinki, Finland and Stanford, California, US) and in these cases the difference between the original version for Lilla Salen and the Ambisonics version is not noticeable. This process integrates well with the idea of speaker agnosticism.

5. CONCLUSIONS AND FURTHER WORKS

In this paper the concept of speaker agnosticism has been contextualised in the field of electroacoustic artistic practice, and the differences between two contrasting ways of thinking about compositional space, organic and Euclidian, has been described. This work focuses on listening as the main drive for the spatial compositional process, independent of the technological and technical aspects of spatialisation, but also on the algorithmic possibilities as derived from this process. However, speaker agnosticism and the organic modality of space can be a new way to approach the development of technologies for spatialisation, as well as helping the 'translation' of electroacoustic pieces between different multichannel speaker systems. Following the results of this work, we see that it would be possible to institute a 'network' of coordinated concert halls that share technical information that provides support to composers and that promotes aesthetic discussion and development in this field. Composers could create 'touring' pieces, where an electroacoustic work composed in Stockholm, for example, could be performed in Leicester maintaining the same spatial musical intent. The introduced pieces, *Travelling without moving* and *For Bill, Rising*, follow this idea: their spatial relevant musical information is kept by using a suitable current technology - Ambisonics, in our case - and by approaching spatialisation from an organic and speaker agnostic perspective.

Acknowledgments

This project has been possible thanks to the Arts and Humanities Research Council (AHRC) and the Midlands4Cities (M4C) doctoral consortium that have funded the author's residency at the Royal College of Music of Stockholm in June-July 2023.

6. REFERENCES

[1] D. Smalley, "Spectromorphology: explaining sound-shapes," *Organised Sound, 2(2)*, pp. 107–126, 1997.

[2] P. Decroupet, E. Ungeheuer, and J. Kohl, "Through the sensory looking-glass: The aesthetic and serial foundations of gesang der jünglinge," *Perspectives of New Music*, pp. 97–142, 1998.

[3] V. Lombardo, F. Nunnari, J. A. Padget, S. Weinzierl, A. Arghinenti, A. Valle, H. H. Vogel, J. Fitch, R. Dobson, J. Padget, K. Tazelaar, S. Benser, S. Kersten, R. Starosolski, W. Borczyk, W. Pytlik, S. Niedbała, and F. Kommunikations-Wissenschaft, "The virtual electronic poem (vep) project," *Proceedings of the International Computer Music Conference, ICMC*, pp. 451–454, 2005. [Online]. Available: http://www.alice-eindhoven.nl

[4] J. W. Solomon, "Spatialization in music: The analysis and interpretation of spatial gestures," Ph.D.

dissertation, 2007. [Online]. Available: http://getd.libs.uga.edu/pdfs/solomon_jason_w_200705_phd.pdf

[5] J. Chowning, "The simulation of moving sound sources," *Computer Music Journal*, vol. 1, 06 1977.

[6] F. Zotter and M. Frank, *A Practical 3D Audio Theory for Recording, Studio Production, Sound Reinforcement, and Virtual Reality*, 1st ed. Springer Cham, 5 2019, vol. 19. [Online]. Available: http://www.springer.com/series/8109

[7] N. Peters, G. Marentakis, and S. McAdams, "Current technologies and compositional practices for spatialization: A qualitative and quantitative analysis," *Computer Music Journal*, vol. 35, pp. 10–27, 3 2011.

[8] B. Blesser and L. Salter, *Spaces Speak, Are You Listening?: Experiencing Aural Architecture*, ser. The MIT Press. MIT Press, 2009.

[9] D. G. Malham and A. Myatt, "3-d sound spatialization using ambisonic techniques," *Computer Music Journal*, vol. 19, no. 4, p. 58, 1995. [Online]. Available: http://dx.doi.org/10.2307/3680991

[10] N. Barrett, "Ambisonics and acousmatic space: a composer's framework for investigating spatial ontology," in *Proceedings of the Sixth Electroacoustic Music Studies Network Conference*, 2010.

[11] C. Travis, "A new mixed-order scheme for ambisonic signals," in *Proc. of the Ambisonics Symposium, Graz, Austria*, 2009.

[12] S. Wilson and J. Harrison, "Rethinking the beast: Recent developments in multichannel composition at birmingham electroacoustic sound theatre," *Organised Sound*, vol. 15, no. 3, p. 239–250, 2010.

[13] G. S. Kendall and M. Ardila, "The artistic play of spatial organization: Spatial attributes, scene analysis and auditory spatial schemata," *Computer Music Modeling and Retrieval. Sense of Sounds, 4th International Symposium, CMMR 2007*, pp. 125–138, 9 2007.

[14] D. Smalley, "Space-form and the acousmatic image," *Organised Sound*, vol. 12, pp. 35–58, 2007.

[15] T. Wishart, *On Sonic Art*, new and rev. ed. ed., S. Emmerson, Ed. Hardwood Academic, 1996.

[16] G. S. Kendall, "Spatial perception and cognition in multichannel audio for electroacoustic music," *Organised Sound*, vol. 15, pp. 228–238, 12 2010.

[17] S. Catena, "Concepts and approaches in analysing spatial sound movements: a link between mozart and acousmatic music," *Proceedings of the XXIII CIM (Colloqui d'Informatica Musicale)*, pp. 105–111, 2022.

[18] U. A. Holbrook, "Sound objects and spatial morphologies," *Organised Sound*, vol. 24, pp. 20–29, 4 2019.

[19] S. R. Quackenbush and J. Herre, "Mpeg standards for compressed representation of immersive audio," *Proceedings of the IEEE*, vol. 109, no. 9, pp. 1578–1589, 2021.

[20] F. Schumacher, V. Espinoza, F. Mardones, R. Vergara, A. Aránguiz, and V. Aguilera, "Perceptual recognition of sound trajectories in space," *Computer Music Journal*, vol. 45, pp. 39–54, 4 2022.

[21] J. Harrison and S. Wilson, "Organised sound: Editorial," *Organised Sound*, vol. 15, pp. 183–184, 12 2010.

[22] J. Chowning, "Turenas: the realization of a dream," 5 2011. [Online]. Available: https://hal.science/hal-03104721

[23] B. L. Jacob, "Algorithmic composition as a model of creativity," *Organised Sound*, vol. 1, no. 3, p. 157–165, 1996.

CSOUND VS. CHUCK: SOUND GENERATION FOR XR MULTI-AGENT AUDIO SYSTEMS IN THE META QUEST 3 USING THE UNITY GAME ENGINE

Pedro LUCAS (pedroplu@ifi.uio.no) [1], **Stefano FASCIANI** (stefano.fasciani@imv.uio.no) [2], and **Kyrre GLETTE** (kyrrehg@ifi.uio.no) [3]

[1, 3] *RITMO, Department of Informatics*, **University of Oslo**, Oslo, Norway
[2] *Department of Musicology*, **University of Oslo**, Oslo, Norway

ABSTRACT

Extended Reality (XR) technologies, particularly headsets like the *Meta Quest 3*, are revolutionizing the field of immersive sound and music applications by offering new depths of user experience. As such, the *Unity* game engine emerges as a preferred platform for building such auditory environments. As part of its capabilities, *Unity* allows the programming of sound generation through a low-level digital signal processing API, which requires specialized knowledge and significant effort for development. However, wrappers that integrate *Unity* with programming languages for sound synthesis can facilitate the implementation of this task. In this work, we focus on applications for the *Meta Quest 3* involving multiple spatialized audio sources; such applications can be framed as XR multi-agent audio systems. We consider two wrappers, *CsoundUnity* and *Chunity*, featuring *Csound* and *ChucK* programming languages. We test and analyze these wrappers in a minimal XR application, varying the number of audio sources to measure the performance of both tools in two device environments: the development machine and the *Meta Quest 3*. We found that *CsoundUnity* performs better in the headset, but *Chunity* performs better in the development machine. We discuss the advantages, limitations, and computational issues found on both wrappers, as well as the criteria for choosing them to develop XR multi-agent audio applications in *Unity*.

1. INTRODUCTION

Currently, *Extended Reality (XR)* technologies are emerging for different aspects of society as part of a novel paradigm. In the field of sound and music, XR technologies offer new possibilities for musicians and audiences, in terms of audio-visual and interactive experiences [1].

The development of XR applications requires platforms capable of covering a range of features to achieve such immersive experiences. Today, two platforms are broadly used to develop these systems: *Unreal* [1] and *Unity* [2]. Both are game engines widely used in the industry. Nevertheless, although they offer an extensive set of tools to develop full XR experiences, their audio capabilities are still insufficient for specific XR sound and music systems [2]. In the context of sound synthesis and analysis, *Unreal* provides a DSP toolset called *MetaSounds* [3]. On the other hand, *Unity* lacks a similar built-in module and requires third-party add-ons. Despite this, Unity is still preferred due to its high compatibility with the XR ecosystem [2].

An option to achieve sound generation in *Unity* is using audio programming languages interfaced through OSC messages [3], which allows the use of widely known languages such as *Max 8*, *Pure Data*, *Faust*, *Csound*, *ChucK*, or others that support OSC connections. However, it implies playing the sound output outside the Unity application, and if we want to have built-in sound generation within the same system, we need to use wrappers that allow direct interfacing with the engine.

In this work, we focus on XR applications developed in Unity for the recently released *Meta Quest 3* XR headset. As the *Meta Quest 3* is based on the Android operative system, the only wrappers that are currently supported for this OS are *CsoundUnity* and *Chunity* interfacing *Csound* and *ChucK* respectively.

Our research aims to investigate whether both wrappers function correctly in Unity when we have multiple spatialized instances of audio sources, which are playing patches developed in these programming languages for an XR application in the *Meta Quest 3*. This scenario often occurs in XR multi-agent audio systems, when several audio sources are present in the same environment. Our study also examines how these wrappers are integrated into the game engine. Furthermore, we analyze their performance in terms of scalability, i.e., their ability to handle an increasing or decreasing number of audio sources, and identify any potential issues in terms of resource utilization.

An additional contribution to this paper is a public GitHub repository [4] that can be used as a base for the development of XR audio systems, and where further testing can be applied.

This paper is organized as follows: Section 2 refers to multi-agent audio systems and the specific technologies that we are using in this work, section 3 describes the

[1] https://www.unrealengine.com
[2] https://unity.com/

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[3] https://docs.unrealengine.com/5.0/en-US/metasounds-in-unreal-engine/
[4] https://github.com/pedro-lucas-bravo/sound_engines

methodology followed in this study, section 4 presents the results and discussion regarding the integration of these wrappers in Unity and the experiments performed over them, section 5 lists recommendations based on the obtained results, and finally in section 6 we provide conclusions and future work.

2. XR MULTI-AGENT AUDIO SYSTEMS

Multi-agent systems are systems that comprise multiple elements called agents. An agent is a computer system that operates within a specific environment and can act autonomously to achieve its objectives [4]. In interactive situations, we can also include humans as additional agents who interact with these artificial agents. On the other hand, systems where sound and music characteristics are emphasized in virtual or physical-virtual immersive environments can be framed in the field of *Music in Extended Realities (Musical XR)*, which is a multidisciplinary area of research focused on XR technologies and experiences with diverse types of projects such as virtual musical instruments, immersive concert experiences, generative musical systems and gamified musical environments [1].

The combination of these two approaches can lead to systems where multiple entities can participate autonomously in sound and music activities or include external interaction of human users as for human-swarm interactive music systems [5] where collective behaviours emerge from the multiple interactions among artificial and living agents. Works as in [6] and [7] can be considered XR multi-agent audio systems that require the integration of several technologies to achieve immersiveness. In some cases, complex integrations are needed due to certain limitations; for instance, in [7], where Unity is used, the sound is generated externally in *Max 8* and therefore requires additional effort for system communication. However, generating sound inside the application would reduce the complexity of communication, and allow us to use the built-in capabilities of the XR devices for spatial audio to provide the sensation of location for the multiple audio sources in an XR scene.

This paper contributes by providing criteria for including sound generation in the latest XR technologies, which we describe in the rest of this section. The focus of this paper is on the programming languages suitable for developing XR multi-agent audio systems using these technologies.

2.1 The *Meta Quest 3*

Meta Quest 3[5] is an XR (VR + MR) headset (shown in Fig. 1) released by *Meta* in October 2023. Its technical specifications are: *Quest system software*, a 64-bit operative system based on Android source code; processor Qualcomm Snapdragon XR2 Gen 2; 8 GB LPDDR5 RAM memory, graphics processor Adreno 740, and 128 GB or 512 GB storage (depending on the version).

Additionally, the device provides spatial audio capabilities through its 2 built-in speakers, which, together with the rest of its features, allows for building immersive sound and music systems contained in one piece of hardware.

[5] https://www.meta.com/quest/quest-3/

Figure 1: *Meta Quest 3* XR headset.

Our objective is to utilize this device also for sound generation, and here we compare different options for its audio programming through Unity.

2.2 The *Unity* Game Engine

Unity[2] is a general-purpose game engine widely used for building real-time systems across several platforms. For our work, we consider the Android platform since the *Meta Quest 3* works under that OS.

In terms of sound generation, Unity offers access to the audio buffer that runs in the audio thread through the `OnAudioFilterRead`[6] function, where sound data is processed independently from other parts of an application in development. In that sense, it is possible to implement complex audio processing or synthesis using the `C#` programming language directly or building custom plug-ins in `C` or `C++`[7]. Such low-level programming can require considerable development effort compared to other high-level programming environments that are designed for interactive sound and media.

For that reason, integrating Unity with a programming language for procedural audio would accelerate the development process and allow a faster exploration of sonic possibilities within the game engine.

2.3 *Csound* and *ChucK* for Unity: *CsoundUnity* and *Chunity*

Csound and *ChucK* are programming languages designed for digital sound synthesis and processing. *Csound* operates as a compiler; hence, the code must be ready beforehand to be compiled and then run to produce sound [8]. *ChucK* is an on-the-fly programming language, which

[6] https://docs.unity3d.com/ScriptReference/MonoBehaviour.OnAudioFilterRead.html

[7] https://docs.unity3d.com/Manual/AudioMixerNativeAudioPlugin.html

[8] https://docs.unity3d.com/Manual/upm-ui.html

[9] https://assetstore.unity.com/

Feature	*CsoundUnity*	*Chunity*
Unity project set up	It uses the *Unity Package Manager* [8] and does not become part of the main project files.	It is downloaded from the *Unity Asset Store* [9] and becomes part of the main project files.
Instantiation	A Unity `GameObject` with an `AudioSource` component attached needs a script called `CsoundUnity` in order to have an instance of *Csound* running. Several instances can be created in the same scene. The destruction of an instance is automatic when the `GameObject` is destroyed.	A Unity `GameObject` with an `AudioSource` component attached needs a script called `ChuckSubInstance`. Additionally, the scene needs to have one `GameObject` with the scripts `TheChuck` and `ChuckMainInstance`, then several instances of `ChuckSubInstance` can be used. The destruction of a `ChuckSubInstance` is explicit when its `GameObject` is destroyed.
Patch Integration	It supports attaching files to the `CsoundUnity` component as well as direct *Csound* code in Unity `C#` scripts.	Direct *ChucK* code can be used through Unity `C#` scripts. Files are supported, but they need to be explicitly managed using the *Chunity* API.
Parameters Access	*Csound* control-rate variables and score events can be manage through `C#` scripts in Unity. Additionally, it is possible to use editor features with graphical controls in the `CsoundUnity` component. It is not possible to communicate from *Csound* to Unity.	*ChucK* control-rate variables, as well as signal and broadcast events, can be managed through `C#` scripts in Unity. It is possible to communicate from *ChucK* to Unity using these types of events.
Language support	Both wrappers support all function generators (opcodes and UGens) of recent releases of *Csound* and *ChucK*; however, it is not ensured that every function generator will work correctly when using on different platforms.	

Table 1: Features for *CsoundUnity* and *Chunity* wrappers.

means that, unlike *Csound*, users can write code while the program is running. It uses a unique time-based concurrent programming model [9].

Both programming languages are interfaced for Unity through *CsoundUnity* [10] and *Chunity* [11] [10], respectively. These wrappers implement the `OnAudioFilterRead` function mentioned previously to achieve sound generation in Unity.

The necessary common features for these wrappers relevant for fulfilling sound generation in Unity are described in Table 1. Additional features can be explored in their corresponding websites[10] [11] .

In section 4.1, we return to these features and discuss additional considerations after integrating these wrappers in an XR Unity application.

3. METHODOLOGY

We initially searched for suitable wrappers that support sound generation in Unity for Android applications, as the *Meta Quest 3* headset is based on that OS. We found that *Csound* and *ChucK* are the current options that offer such functionality through the wrappers *CsoundUnity* and *Chunity*, respectively. Both programming languages and their wrappers are actively maintained.

A Unity project for the Android platform was set up and configured for the *Meta Quest 3* device. Both wrappers were added to this project. Then, we implemented a simple scene in *Pass-through* mode (i.e. mixed reality or physical-virtual environment) as shown in Fig. 2, consisting of coloured spheres rotating around a cube that can be moved by the user using any hand. Every sphere has an audio source attached, which is an instance of a continuous sound built through one of the wrappers. A sphere can be spawned in a random location around the perimeter of the cube and adopt a random colour. With this simple scene, we wanted to assess that spatial audio works for both wrappers by hearing the output as we moved in the scene. A video showing this scene in action was published [12] . This video was captured using a streaming service for *Meta Quest 3* [13] with the limitation of having monaural audio.

We used the scene to perform a series of experiments when the number of sources was varied to collect performance measurements. These experiments were executed based on the following scenarios:

1. **Scalability:** In this scenario, we start the scene without sound sources, and then we add one by one. With every increase, we take several measurements of the digital signal processing (DSP) time that Unity spends on generating the sound in its audio buffer. Additionally, we measure the memory that Unity uses exclusively for audio. Both parameters, *DSP time* and *audio memory*, are frequently sampled dur-

[10] `https://github.com/rorywalsh/CsoundUnity`
[11] `https://chuck.cs.princeton.edu/chunity/`
[12] `https://youtu.be/OtpDR9MnnbI`
[13] `https://www.oculus.com/casting`

Figure 2: Screenshot for the mixed reality application taken in the *Meta Quest 3*. The audio sources are represented by coloured spheres orbiting around an interactive cube that can be moved using any user's hand. In this scene, 29 objects were instantiated, but some are not visible due to the field of view limit.

ing a period of time before adding a new source. While this experiment is running, we listen to the output to identify the number of sources that produce audio artifacts (e.g. glitches), indicating that sound processing is taking longer than allowed for real-time.

2. **Resource Leak Detection:** For this case, we start again without any sources, then we take several measurements for *DSP time* and *audio memory* in a determined period; next we add one audio source that will be playing for some time, and then we remove this source and take measurements again without any sources. We repeat this trial several times. The goal is to identify whether the DSP time and the audio memory are the same as we increase the number of trials. The increase in any of both parameters would indicate resource leak problems.

To collect these measurements, we implemented a custom script using the `ProfilerRecorder`[14] API from Unity to access the *audio memory*. For the *DSP time*, we measured the execution time for the code inside the `OnAudioFilterRead` function.

We applied both experiment scenarios to the following two devices:

1. **The Development Machine:** This is the environment where the XR application is developed. The experiments are performed over the application running in the Unity editor. The machine used was a Windows 11, 64-bit, powered by a 12th Gen Intel(R) Core(TM) i7-12700H 2.30 GHz, with RAM 16 GB, and a laptop graphics processor NVIDIA GeForce RTX 3050 Ti.

2. **The *Meta Quest 3*:** A XR headset that supports virtual and mixed reality applications. The device used was the version with 128 GB of storage. Its specifications are described in section 2.1.

[14] https://docs.unity3d.com/ScriptReference/Unity.Profiling.ProfilerRecorder.html

For the first scenario, we considered implementing two patches: a *sine oscillator*, and a *simple synthesizer* having the signal chain `saw_oscillator` → `low_pass_filter` → `reverb` → `amplitude_envelope` → `gain`. The second scenario used only the sine oscillator patch.

An audio source that uses the sine oscillator plays continuously, and a random frequency is assigned when it is instantiated in the scene. For the simple synthesizer, the envelope plays one second every 2 seconds, and parameters for the components in the chain are changed randomly on every play. Implementation details can be explored in our public repository[4] .

For all the experiments, we set the *DSP buffer size* in Unity as `Best Performance`, meaning a buffer size of 1024 samples. We used the default audio sample rate in Unity (48000 Hz), thus the buffering latency of real-time audio is 21.33 ms. The specific settings for every experiment, as well as results and interpretations, are presented in section 4.2.

4. RESULTS AND DISCUSSION

4.1 Integration with Unity

After the testing performed, we can confirm that both wrappers are operational on the *Meta Quest 3* for the development of XR audio applications. However, there are differences in the development process that might influence the decision to use one or both of them. We discuss some of these differences as follows:

- **Learning the languages:** *Csound* and *ChucK* use very different syntax. While *ChucK* utilizes a more traditional syntax similar to programming languages like `Java`, `JavaScript`, and `C#`, making it relatively easy to learn, *Csound* has a unique syntax that may require some effort to learn, especially for those who are inexperienced. However, the effort is worth it if we want to take advantage of the many function generators offered by *Csound*, which might not be available in *ChucK*.

- **Implementing the patches:** The complexity depends on what a developer wants to achieve and what the languages offer. For instance, a simple sine wave could be relatively straightforward in both languages; however, other implementations, such as an amplitude envelope, can differ depending on the language paradigm. In our case, we implemented such an envelope for a simple synthesizer. Differences can be identified in the Unity scripts for the synthesizer behaviour[15] [16] . Additionally, *Csound* offers more built-in function generators than *ChucK*; nevertheless, *ChucK* allows to create custom plug-

[15] https://github.com/pedro-lucas-bravo/sound_engines/blob/main/sound_engines_unity_project/Assets/Scripts/CsoundSimpleSynthe.cs

[16] https://github.com/pedro-lucas-bravo/sound_engines/blob/main/sound_engines_unity_project/Assets/Scripts/ChuckSimpleSynthe.cs

ins and expand the possibilities for signal processing [17].

- **Setting up a Unity project:** In terms of project organization and file optimization, *CsoundUnity* has an advantage over *Chunity*, since it can be included in a Unity project through the *Unity Package Manager*[8] allowing its access locally in the development machine and saving space for a version control system (e.g. git) since the package is downloaded every time the project is cloned. As *Chunity* files lie within the project itself, they must be tracked by the version control system.

- **Developing multi-agent audio systems:** Instantiation and destruction of audio sources occur in a multi-agent audio system and are carried out according to the application's needs. We would expect to create an audio agent with a particular behaviour that is independent of others and not implement additional considerations to ensure that independence. In the case of *CsoundUnity*, this independence is ensured for every new instance created; however, *Chunity* needs an extra effort to achieve it; that is, variables in a *ChucK* script must be different for every instance, thus if we are using the same patch, we have to rename the variables to not interfere among sources, which requires additional code to automatize this process. Due to this issue and in comparison to *Chunity*, *CsoundUnity* facilitates the development of a multi-agent audio system.

- **The system in action:** Perceptually, there is no difference between the output generated by the wrappers, which means that any of them can be used for the intended purpose. Moreover, there were no crashes or artifacts observed when a reasonable number of sources were used in the experiments. However, there can be differences in performance, as explained in the next section.

Furthermore, it is possible to include in one project both wrappers at the same time, which increases the options for sound generation in a *Meta Quest 3* application.

4.2 Computational Performance

We accomplished the experiment scenarios specified in section 3 on the simple scene from the mixed reality application mentioned in the same section. We initially used the *Unity Profiler* [18] to look for noticeable performance issues like frame-rate drops or important memory leaks. We did not find such issues at that stage for both wrappers. Then, we executed the experiment scenarios with the following results.

4.2.1 Scalability

For the scalability scenario, we added up to 30 audio sources for the *sine oscillator* and the *simple synthesizer*

[17] https://chuck.cs.princeton.edu/extend/
[18] https://docs.unity3d.com/Manual/Profiler.html

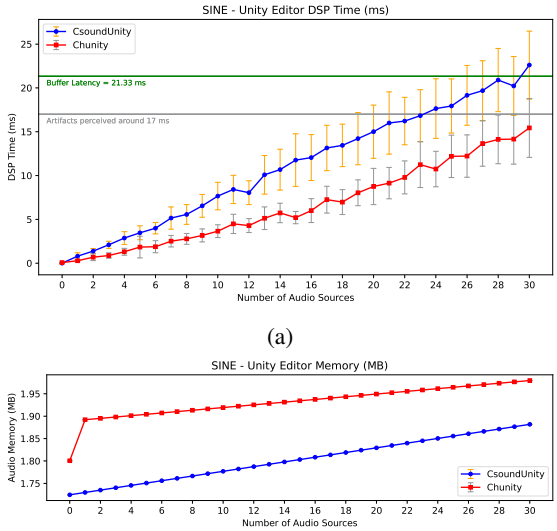
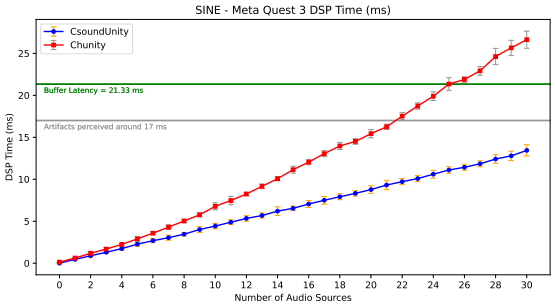


Figure 3: Results for the *Sine Oscillator* patch tested in the Unity Editor (development machine) (a) DSP Time. (b) Audio Memory.

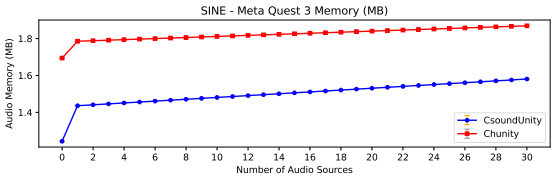
patch. For both patches we started from zero sources, then we took measurements every 0.2 seconds until having 30 samples; next, we rested for 1 second before adding a new audio source, and so on. We performed this procedure for the Unity Editor (development machine) and the *Meta Quest 3*. Fig. 3 shows the results for the Unity Editor and Fig. 4 for the *Meta Quest 3* regarding the DSP time and the audio memory for the sine oscillator patch. The charts depict the average value and standard deviation for each number of audio sources. Note that every chart illustrates the buffer latency (21.33 ms), although we experimentally realized that glitches were perceived around 17 ms since Unity needs to use additional computational time for internal processes.

The sine oscillator patch: Note that, for the Unity Editor in Fig. 3, *Chunity* invests less DSP time than *CsoundUnity*; however, *Chunity* consumes more audio memory, although this amount is relatively low (less than 2 MB) for the overall system memory. Considering the threshold for artifacts perception of 17 ms, we found that glitches start to appear for *CsoundUnity* when there are around 25 sources, and for *Chunity* when we have 37. Although these results may vary according to the development machine, the important aspect is that ***Chunity* performs better than *CsoundUnity* in the Unity Editor under the Windows implementation of *ChucK* and *Csound***. Memory usage can be negligible in this case.

For the *Meta Quest 3* in Fig. 4, the results are different from the Unity Editor since we have less variability in the data, which is expected because the editor deals with additional processes. Moreover, the most important difference is that *CsoundUnity* takes less DSP time than *Chunity*, which is the opposite tendency regarding the Unity Editor. For the audio memory, we still have a negligible result under 2 MB. Therefore ***CsoundUnity* performs better than *Chunity* in the *Meta Quest 3* under the Android imple-**



(a)



(b)

Figure 4: Results for the *Sine Oscillator* patch tested in the *Meta Quest 3*. (a) DSP Time. (b) Audio Memory.

mentation of *ChucK* and *Csound*. We start to perceive glitches for *CsoundUnity* when we have 39 sources, and for *Chunity* when there are 22.

The simple synthesizer patch: A more realistic example was tested considering the simple synthesizer mentioned in section 3. We show the results for the Unity Editor and the *Meta Quest 3* in Fig. 5 and Fig. 6, respectively. In this case, for both devices, the audio memory usage is similar to the sine oscillator patch and thus is not necessary to show these results; however, we can notice important differences in the DSP time. These results support the previous statements regarding which wrapper is better according to the device, although we notice that *Chunity* takes less DSP time for this patch than the sine oscillator in such a way that its performance is similar in the *Meta Quest 3* until 18 sources in Fig. 6. We speculate that *Chunity* performs better in this case due to how *ChucK* manages the amplitude envelope in comparison to *Csound* and due to the resting period between the envelope execution since it is played for one second every 2 seconds on each audio source. Nevertheless, we cannot ensure a fair comparison for this patch since we do not know details about the internal implementation of the function generators for each programming language; hence, results might change if using different generators (e.g. using a `moogladder` low pass filter instead of a `lowres` in *Csound*).

4.2.2 Resource Leak Detection

We performed the second scenario described in section 3 to detect resource leaks in DSP time and audio memory for both devices using the sine oscillator patch. We perform 100 trials for each device. Fig. 7 and Fig. 8 show the results for the Unity Editor and the *Meta Quest 3*, respectively. Ideally, we would expect zero DSP time process and a constant memory that is not invested in the audio generation after several adding-removing sources actions; however, we detect the following issues:

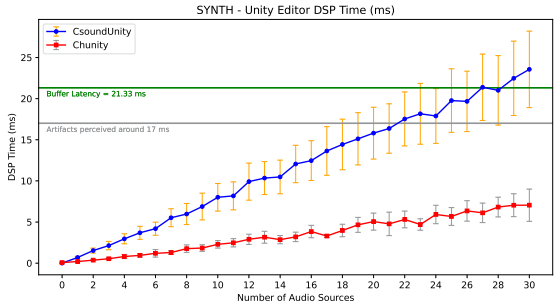


Figure 5: DSP time results for the *Simple Synthesizer* patch tested in the Unity Editor (development machine).

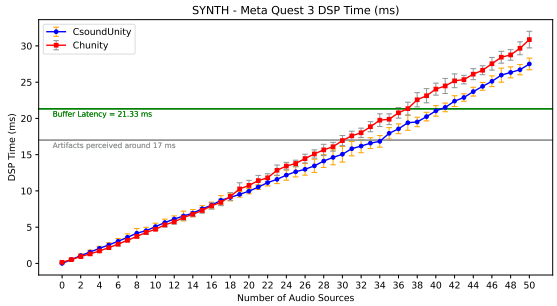
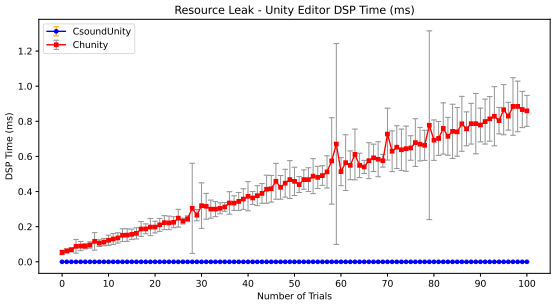


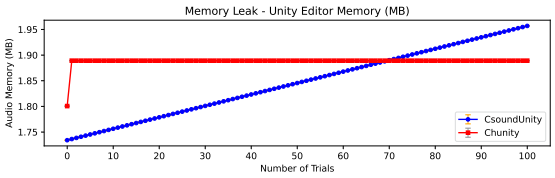
Figure 6: DSP time results for the *Simple Synthesizer* patch tested in the *Meta Quest 3*.

Resource Leak for DSP Time: In both devices, we verify that *CsoundUnity* unity is not processing anything during the several trials in both devices; that is, it portrays the expected behaviour. Nevertheless, *Chunity* starts with a small DSP time even though we have not started any adding-removing process, and then this time increases linearly as we progress in the trials for the two platforms. After the 100 trials, *Chunity* wasted close to 1 ms in resources for the Unity Editor, as shown in Fig. 7a, while for the *Meta Quest 3* in Fig. 8a, this value is close to 3 ms, although the variability is lower than in the editor. We hypothesize that the reason behind this issue is that *Chunity* would manage a centralized execution of *ChucK* scripts since every variable must be different per instance, and when we stop or remove an audio source, the code is still running even though the sound generation is not happening anymore. Moreover, *Chunity* needs the `TheChuck` and `ChuckMainInstance` components as pre-requisites to have multiple audio sources in one scene, which may cause the initial processing time when there are no sources yet.

Memory Leak: In the case of audio memory, we identify an initial increment for *Chunity* in both devices as shown in Fig. 7b and Fig. 8b. For the Unity Editor, *Chunity* reaches a value close to 1.9 MB that is constant along the trials, while for the *Meta Quest 3*, this value is slightly higher than 1.8 MB. In that sense, *Chunity* fulfils the expectation of constant memory. However, *CsoundUnity* denotes a linear increment of memory that overcomes *Chunity* in the editor as shown in Fig. 7b; that is, it passes 1.9 MB around the trial 70. For the *Meta Quest 3*, the audio memory for *CsoundUnity* is also increasing but, after the 100 trials, is still below the 1.8 MB reached by *Chu-*

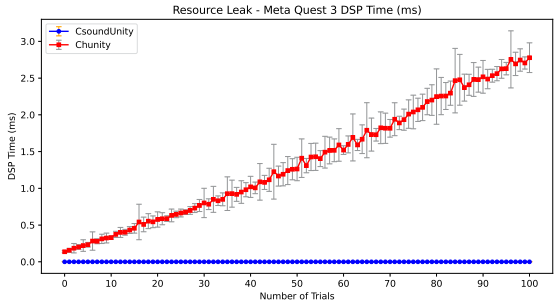


(a)

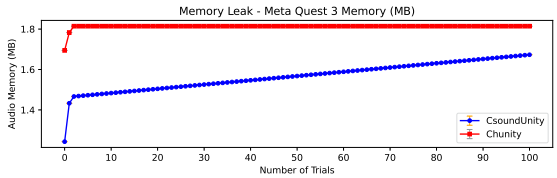


(b)

Figure 7: Resource Leak Detection in the Unity Editor (development machine) (a) DSP Time. (b) Audio Memory.



(a)



(b)

Figure 8: Resource Leak Detection in the *Meta Quest 3*. (a) DSP Time. (b) Audio Memory.

nity. We think that, although *CsoundUnity* is not processing audio anymore, the implementation is not freeing all the memory and, therefore, leading to this leak. While, for *Chunity*, although it is still processing and spending unnecessary DSP time, the audio memory does not increase since the audio generation is stopped, although it could be using the main memory to keep track of the hanging scripts while adding and removing sources.

5. RECOMMENDATIONS

Based on the results and their interpretations provided in the previous section, we list the following recommendations for developing multi-agent audio systems in the *Meta Quest 3*.

- **Develop for the target platform:** For our case, the target platform is the *Meta Quest 3*. If choosing a wrapper does not depend on factors other than performance, we can go for *CsoundUnity* according to our results; however, other considerations may influence this decision depending on the system, time restrictions, and the developer's skills. The important aspect is that the Unity Editor should not be used as an accurate reference for how an XR system performs.
- **Detection of issues through profiling tools:** For the *Meta Quest 3* we can use the *Unity Profiler*[18] and the *Meta Quest Developer Hub*[19] as debugging tools for quick identification of performance issues. Custom profiling tools can be implemented as in this work. Use these options to improve the application's performance if critical issues arise.
- **Estimate the maximum number of agents:** As the results show, both wrappers present resource leak issues. A way to work around this problem, and as a good practice in real-time systems, is to use *object pools* and just active or deactivate objects. It can be useful not only for audio but for other components when complete agents are pooled.
- **Identify additional audio processes:** An XR system may use additional audio objects that increase the DSP time in Unity (e.g. mixer, plugins, etc). Being aware of these additional processes, which can be quantified through profiling, is important to avoid overloading the audio buffer.
- **Think in multi-platform if needed :** Unity is a multi-platform game engine, so the same application may be potentially built for different devices and operative systems. As we have shown in the results, performance can vary depending on the platform (Windows in the editor and Android in the *Meta Quest 3*). If the development requires targeting several platforms, we need to choose the wrapper that best fits all of them, and still, the ideal case is to focus on one target platform, as pointed out before.
- **Optimization beyond Unity :***Csound* and *ChucK* have particular programming paradigms and their own ways of optimization. An example is the close performance for both in the simple synthesizer experiment, as shown previously in the results. Use the power of any of those languages as much as possible to achieve highly optimized sound generation that helps to reach your performance goals for an XR application in Unity.
- **Explore choices according to the system :** We might encounter various challenges while developing a system. Depending on the particular requirements, we may need to change our decision regarding which engine to use. For instance, if you are

[19] https://developer.oculus.com/meta-quest-developer-hub/

more fluent in *ChucK* or *Csound*, and performance is not critical to a certain extent, then you can use any or even both languages. However, you should consider the earlier recommendations and balance your choices based on your skills.

6. CONCLUSIONS

We explored *Csound* and *ChucK* as mediums for sound generation in Unity, focused on multi-agent audio systems running in the *Meta Quest 3* through the wrappers *CsoundUnity* and *Chunity*. We developed a simple mixed reality scene to test both wrappers and discussed their integration into Unity. The same scene was used for several experiments to evaluate their computational performance regarding DSP time and audio memory. We confirmed that both wrappers are effective with spatialization in the *Meta Quest 3*.

In terms of integration with Unity, *CsoundUnity* ensures independence for audio sources that implement the same behaviour, while *Chunity* needs explicit identification of every source, which requires an extra development effort. Additionally, both wrappers can be used for the same application.

In terms of computational performance, results regarding the DSP time tell us that *Chunity* performs better than *CsoundUnity* in the Unity Editor for a Windows machine, but *Csound* is more efficient in the *Meta Quest 3* (based on Android). When using any of the wrappers, the audio memory is relatively low (less than 2 MB) and thus negligible to have a considerable impact. Despite these results, patches might be more or less efficient depending on how they are implemented and how efficient the function generators are in every language, which might influence in the results.

We discovered resource leak issues for both wrappers. *Chunity* consumes DSP time when several adding-removing actions happen, while *CsoundUnity* does not free all memory under the same conditions and *Chunity* does; however, *CsoundUnity* does not consume DSP Time. An object pool can be used to mitigate these issues.

Based on these results, we suggested the recommendations listed in the previous section that can be used to select the most suitable wrapper (or both) for an XR application.

For future work, we plan to assess these wrappers for other platforms supported by Unity, such as general Android devices, iOS devices, PCs (Mac, Linux, Windows), and web browsers. In addition, we aim to enable an XR application to switch between the default spatial audio system for the *Meta Quest 3* and an external multi-speaker system. To achieve this, we need to investigate how to use *Csound* and *ChucK* patches in external sound engines that support ambisonic capabilities.

Acknowledgments

This work was partially supported by the Research Council of Norway through its Centres of Excellence scheme, project number 262762.

7. REFERENCES

[1] L. Turchet, R. Hamilton, and A. Camci, "Music in Extended Realities," *IEEE Access*, vol. 9, pp. 15 810–15 832, 2021. [Online]. Available: https://ieeexplore.ieee.org/document/9328440/

[2] L. Turchet, "Musical Metaverse: vision, opportunities, and challenges," *Personal and Ubiquitous Computing*, vol. 27, no. 5, pp. 1811–1827, Oct. 2023. [Online]. Available: https://link.springer.com/10.1007/s00779-023-01708-1

[3] D. Johnson, D. Damian, and G. Tzanetakis, "OSC-XR: A Toolkit for Extended Reality Immersive Music Interfaces," in *16th Sound and Music Computing Conference*, Málaga, Spain, May 2019, iSBN: 9788409085187 Publisher: Zenodo. [Online]. Available: https://zenodo.org/record/3249318

[4] M. Wooldridge, *An Introduction to MultiAgent Systems*, 2nd ed. Wiley Publishing, 2009.

[5] P. Lucas and K. Glette, "Human-Swarm Interactive Music Systems: Design, Algorithms, Technologies, and Evaluation," *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research*, Nov. 2023, publisher: Zenodo. [Online]. Available: https://zenodo.org/doi/10.5281/zenodo.10113080

[6] A. Bargum, O. I. Kristjansson, S. Rostami Mosen, and S. Serafin, "Spatial Audio Mixing in Virtual Reality," in *19th Sound and Music Computing Conference*, Saint-Étienne, France, Jun. 2022, publisher: Zenodo. [Online]. Available: https://zenodo.org/record/6572821

[7] P. P. Lucas and S. Fasciani, "A Human-Agents Music Performance System in an Extended Reality Environment," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, M. Ortiz and A. Marquez-Borbon, Eds., May 2023, pp. 10–20, place: Mexico City, Mexico. [Online]. Available: http://nime.org/proceedings/2023/nime2023_2.pdf

[8] V. Lazzarini, O. Brandtsegg, J. ffitch, J. Heintz, I. McCurdy, and S. Yi, *Csound: A Sound and Music Computing System*, 1st ed. Cham: Springer International Publishing : Imprint: Springer, 2016.

[9] G. Wang, P. R. Cook, and S. Salazar, "ChucK: A Strongly Timed Computer Music Language," *Computer Music Journal*, vol. 39, no. 4, pp. 10–29, Dec. 2015. [Online]. Available: https://direct.mit.edu/comj/article/39/4/10/106778/ChucK-A-Strongly-Timed-Computer-Music-Language

[10] J. Atherton and G. Wang, "Chunity: Integrated Audiovisual Programming in Unity," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, T. M. Luke Dahl, Douglas Bowman, Ed. Blacksburg, Virginia, USA: Virginia Tech, Jun. 2018, pp. 102–107, iSSN: 2220-4806. [Online]. Available: http://www.nime.org/proceedings/2018/nime2018_paper0024.pdf

COMPARING FOUR 360-DEGREE CAMERAS FOR SPATIAL VIDEO RECORDING AND ANALYSIS

Jinyue GUO (郭锦岳) (jinyueg@uio.no) [1,2], **Maham RIAZ** (mahamr@uio.no) [1,2], and **Alexander R. JENSENIUS** (alexanje@uio.no) [1,2]

[1] *RITMO Centre for Interdisciplinary Studies in Rhythm, Time and Motion*, **University of Oslo**, Oslo, Norway
[2] *Department of Musicology*, **University of Oslo**, Oslo, Norway

ABSTRACT

This paper reports on a desktop investigation and a lab experiment comparing the video recording capabilities of four commercially available 360-degree cameras: GoPro MAX, Insta360 X3, Garmin VIRB 360, and Ricoh Theta S. The four cameras all use different recording formats and settings and have varying video quality and software support. This makes it difficult to conduct analyses and compare between devices. We have implemented new functions in the Musical Gestures Toolbox (MGT) for reading and merging files from the different platforms. Using the capabilities of FFmpeg, we have also made a new function for converting between different 360-degree video projections and formats. This allows (music) researchers to exploit 360-degree video recordings using regular video-based analysis pipelines.

1. INTRODUCTION

So-called 360-degree video—often also called spherical video—is popular in immersive content creation but is still used relatively little in research. We are exploring 360-degree video recordings for various sound and music projects where sound sources are moving or distributed across an environment. This includes music performances in concert halls, where a 360-degree camera can capture both musicians and the audience. It also includes studies where dancers move around a space. Our recent research is focused on the audiovisual qualities of in-door environments, where capturing the entire scene is essential. Regular video recordings only capture a small portion of space, while 360-degree cameras can capture the entire scene if positioned well.

Even though 360-degree cameras have been on the market for some years, they are still a niche product, and the manufacturers have not agreed on standard settings and formats. Few software solutions are available for editing, and to our knowledge, no analysis tools are available. In sum, while there is a huge potential for using 360-degree video in research, the existing commercial 360 cameras and their software have insufficient support for ad-

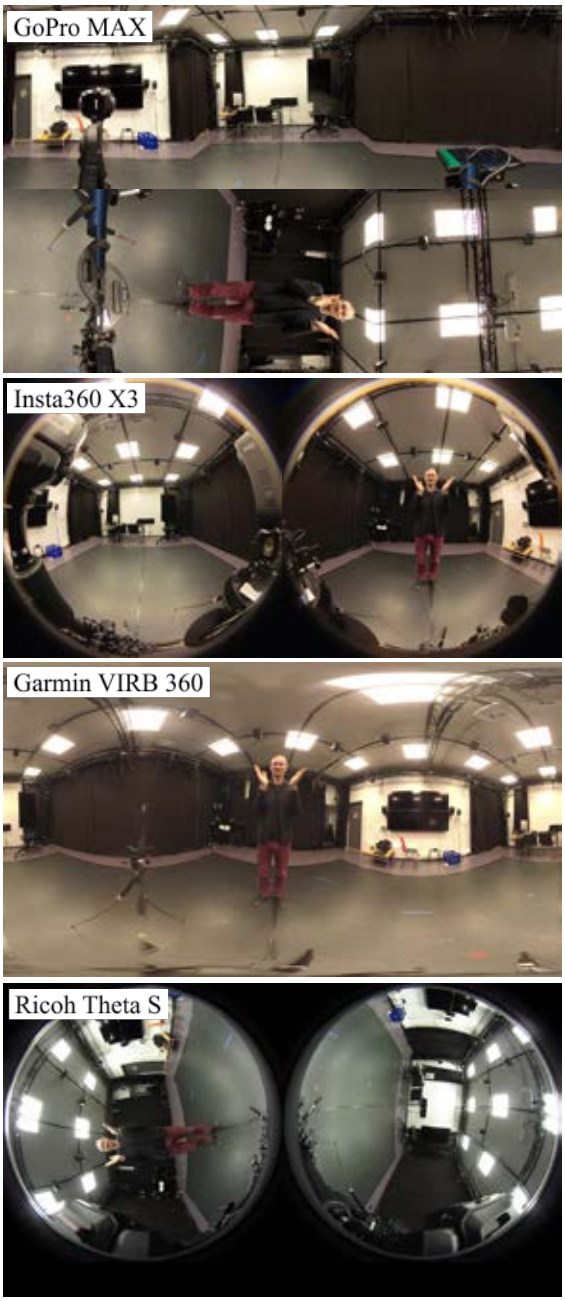


Figure 1. Original projections of the four cameras. GoPro MAX: modified equi-angle cubemap; Insta360 X3: dual-fisheye in two separated files; Garmin VIRB 360: equirectangular; Ricoh Theta S: dual-fisheye in one file.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

vanced users and researchers. In this paper, we review four commercially available 360-degree cameras, describe their recording specifications and formats, and present how we implemented functions in the Musical Gestures Toolbox [1] for analyzing the recorded files.

2. ABOUT 360-DEGREE VIDEO

360-degree video is a visual media type that provides an immersive viewing experience. Such videos refer to omnidirectional vision, which entails a full spherical view, including 360 degrees in both horizontal and vertical directions.

Immersive recording and production techniques are becoming popular in the gaming and film industries. In broadcasting, curved monitors and television screens lead the way with an extended field of view. Head-mounted displays (HMDs), like virtual and mixed reality headsets, feature head tracking, enabling dynamic perspective adjustment with natural head motion, thus offering viewers a full 360-degree experience.

On the production end, new techniques have revolutionized set design, with the virtual reconstruction of entire filming locations, using circular screens spanning up to 270 degrees [1]. A 360-degree video is a cost-effective option for immersive recordings. This advanced visual medium has become popular with filmmakers, videographers, athletes, and tech enthusiasts looking for user-friendly recording equipment.

360-degree video recordings can manifest in several representations. Most commercial 360-degree cameras consist of two or more fish-eye cameras and contain internal stitching and processing algorithms that save the videos in a processed format. The saved files differ hugely between models, but most manufacturers provide editing software that allows users to perform post-processing, including trimming, stabilization, denoising, and projection conversion. This makes it possible to render 2D video files for "normal" video editing.

The processing pipeline of 360-degree videos differs from traditional videos mostly due to its spherical property [2]. It contains four steps [3]:

1. Stitching: The video is often captured using multiple cameras, resulting in multiple image views that must be combined ("stitched").

2. Projection: The 360-degree video needs to be projected to a surface and then encoded with compression algorithms for storage.

3. Encoding: The video must then be encoded and transmitted to the encoded media using traditional or customized protocols.

4. Rendering: Finally, the file is decoded and rendered on the viewer's device.

Unlike traditional video processing, the steps of a 360-degree video pipeline cannot be separated from each other [2]. The stitching algorithm will affect the projection, the efficiency of compression methods will change with different projections, and the transmission is adaptive to the viewer's device and viewing habits.

[1] https://github.com/fourMs/MGT-python/

Figure 2. The experimental setup: four 360-degree cameras (left to right: GoPro MAX, Insta360 X3, Garmin VIRB 360 and Ricoh Theta S). A Samsung Galaxy S23 Ulta mobile phone captured sound and light levels.

Projection is an important step in the pipeline since all the following steps need to be adjusted according to the projection [2]. For example, the *equirectangular projection* (ERP) [4] is widely adopted and equally projects the spherical video from its center to a flat surface. The projection largely expands the pole areas that are not likely viewed by the viewers, therefore the compression algorithm can utilize a lower bitrate at the upper and lower edge of the equirectangular projection video. The transmission algorithm can also prioritize the middle section around the equator since it is more likely to be viewed.

3. FOUR 360-DEGREE CAMERAS

We have been using 360-degree cameras in our lab for around a decade and decided to conduct a comparative study between four commercially available devices. Four 360-degree cameras were placed next to each other at the same height ($\sim$1 meter), and a mobile phone recorded the "ground truth" light intensity in the lab (Figure 2).

In the following, we will only investigate the video part of the recorded files; the audio part is the subject of another paper [5]. The recordings were done with the best possible settings in each camera. All of them typically allow for reducing the pixel dimensions or frame rates, but we wanted to check the maximum capabilities of each camera.

Each camera has its own software solution for handling the recordings. We investigated the original files saved by each camera, as well as the export options in their corresponding software.

Camera	File type	Projection	Codec	Colorspace	Resolution	FPS	Bitrate (kb/s)
GoPro MAX	.360	Equi-Angular Cubemap	H.265	yuvj420p	4096 x (2688)	25	30,002 (x2)
	.LRV	Dual-fisheye	H.264	yuvj420p	1408 x 704	25	2499
Insta360 X3	.INSV	Fisheye (x2)	H.264	yuvj420p	(5760) x 2880	29.97	60,495 (x2)
	.LRV	Dual-fisheye	H.264	yuvj420p	1024 x 512	29.97	3999
Garmin VIRB 360	.MP4	Equirectangular	H.264	yuv420p	3840 x 2160	25	80,008
	.LRV	Equirectangular	H.264	yuv420p	1280 x 720	25	5026
Ricoh Theta S	.MP4	Dual-fisheye	H.264	yuvj420p	1920 x 1080	29.97	15,938

Table 1. A comparison of the original files from four different 360-degree cameras.

3.1 GoPro MAX

The GoPro MAX records in 3K resolution at 50 frames per second (fps) or 5.6K resolution at 25 fps. We chose the latter in this experiment since spatial resolution is more important than temporal in our ongoing project. The GoPro MAX produces many files for each recording. Three file types are stored: a high-resolution file (.360), a low-resolution file (.LRV), and a thumbnail image of the first video frame (.THM). For compatibility reasons, GoPro stores the .360 files in chunks of size 4.0 GB (the limitation of FAT32-formatted drives). It also stores the .LRV files in chunks of similar duration as the .360 files, even though that is unnecessary from a file format perspective.

Each .360 file contains 7 data streams: two video streams, two audio streams, and three data streams. The two video streams contain data that is similar to Google's Equal Area Cubemap (EAC) [2], but optimized to squeeze in as many pixels as possible [3]. The data stream with handler name "GoPro MET" contains metadata in GoPro's open source metadata format GPMF [4]. The "GoPro TCD" stream contains a starting timecode, representing the time since midnight of each frame, and the "GoPro SOS" data stream contains data for file recovery.

The GoPro MAX comes with the GoPro Player software that provides basic editing and export functionalities. The software allows users to edit viewing angles as key frames and export a reframed video, but there is no option to export the video in full 360 degrees. It also contains paid functionalities, such as advanced stabilization algorithms that require a subscription. Apart from the GoPro Player, the company also provides an open-source GPMF-parser [5] that can extract the metadata in their own GoPro Metadata Format. However, the GoPro Player does not have open-source code for stitching and converting their customized Equal Area Cubemap projection into common public formats such as Equirectangular Projection, which makes it hard for advanced users and researchers to process their videos with open-source tools such as FFmpeg.

[2] https://blog.google/products/google-ar-vr/bringing-pixels-front-and-center-vr-video/
[3] https://gopro.com/en/us/news/max-tech-specs-stitching-resolution
[4] https://gopro.com/en/sg/news/gopro-video-metadata-open-source-explained
[5] https://github.com/gopro/gpmf-parser

3.2 Insta360 X3

The Insta360 was set to record in 5.7K resolution at 29.97 fps. It saves two types of video files: two high-resolution single-fisheye files (.INSV) and a low-resolution dual-fisheye file (.LRV). Each file is only 1 minute long, which makes for a lot of files for longer recordings. This design choice may be smart from an action camera perspective, where crashes would not lead to corrupt files, but it is annoying with numerous recorded files for longer recordings. Both .LRV and .INSV files contain only two streams, one video and one audio. There are no metadata streams.

The Insta360 Studio 2024 contains various editing functionalities. Apart from basic trimming and editing, it also supports stabilization, motion ND, object tracking, and color filtering. It is the only software that supports various stitching configurations, including dynamic stitching, optical flow-based stitching, chromatic calibration, etc. Users can choose between a reframed export or a full 360-degree export.

3.3 Garmin VIRB 360

The Garmin VIRB 360 is a legacy product, but it has been a popular camera for many years, and we have used it for numerous concert recordings. It offers 360-degree recordings in 4K, automatically stitched by the camera, and RAW 5.7K recordings, which the user can stitch together using proprietary software. The RAW option records and outputs two separate 200-degree hemispherical videos from the lenses as separate files, with a total resolution of 5.7K. In this experiment, the camera was set to record in 360 at 4K. The manual and packaging of the Garmin VIRB 360 states that it records in 30 fps [6]. However, this functionality is preceded by the camera's regional video format settings (whether set to PAL or NTSC). In Europe, PAL is the standard for video recordings. This sets the frame rate to 25fps by default.

The camera saves three files for each recording: one high-resolution equirectangular recording (.MP4), one low-resolution equirectangular recording (.GLV), and a thumbnail image (.THM). Both the .MP4 and .GLV files contain one video stream and one audio stream. There are no metadata streams.

[6] https://static.garmin.com/pumac/Virb_360_OM-EN.pdf

Camera	Software	Export projection	Export view range	Codec	Export resolution	FPS	Comment
GoPro MAX	GoPro Player	ERP	reframed,full 360	H.265, H.264, ProRes	up to 5376 x 2688 (5.6K)	25	
	GPMF-parser	-	-	-	-	25	exports metadata
Insta360 X3	Insta360 Studio	ERP	reframed, full 360	H.265, H.264, ProRes	1920 x 1080 (reframed), 5760 x 2880 (full 360)	29.97	
Garmin VIRB 360	Garmin VIRB Edit	ERP	reframed, full 360	H.264	1920 x 1080 (reframed), 3840 x 2160 (full 360)	25	
Ricoh Theta S	Ricoh THETA	ERP	full 360	H.264	1920 x 960	29.97	no editing functions

Table 2. A comparison of the export functionalities of different software. EAC stands for Equi-Angular Cubemap, and ERP stands for Equirectangular mapping.

The Garmin VIRB Edit software provides basic editing functionalities, including trimming and adjusting the viewing angle. The software supports the G-Metrix metadata format for more functionalities, such as motion detection and GPS localization. However, the Garmin VIRB 360 camera does not provide any G-Metrix data in its recordings, so this option may be related to compatibility with other hardware devices from the same manufacturer. As for export options, the software supports both reframed and full 360-degree export.

3.4 Ricoh Theta S

The Ricoh Theta S is the oldest camera in our selection and offers the fewest options. It only records one 1080p, dual-fisheye .MP4 file with a single video stream. As such, it is the easiest camera to work with from a file format perspective. However, it uses an unusual dual fisheye projection, with each spherical view shifted 90 degrees.

The Ricoh THETA editing software is basic, only supporting simple playback and conversion functionality. It does not offer editing options but only allows the user to import a single dual-fisheye file and export it to a 360-degree file with an equirectangular projection.

4. VIDEO QUALITY COMPARISON

Since 360-degree videos have a long processing pipeline, different distortions might be introduced, even if we do not consider the intrinsic differences between the cameras. Moreover, the distortions vary drastically for different camera models, and these differences are usually not mentioned in their user manuals. These differences might cause undesired analysis errors if researchers compare camera recordings. In this section, we will compare the distortions and the quality of different camera models so that researchers can avoid some of the errors in their experiments.

4.1 Stitching distortion

The recordings were compared using equirectangular projection to understand the stitching distortion in the cameras' output. A common effect in 360-degree cameras is the distortion of objects in the near field. All our cameras have two lenses: one in the front, the other in the back. The field-of-view (FOV) of one lens in any of the cameras in this experiment is unknown. Indeed, any FOV value exceeding 180 degrees will inevitably result in a parallax error at certain points in the combined image. The lenses cannot physically be precisely at the same point, just like human binocular vision. Both eyes have slightly different viewing angles; focusing on an object far ahead produces a clear image, but as the object draws near, the image gets distorted [6]. In equirectangular projections from 360-degree videos, this distortion manifests at the stitching points.

Similar to how the human nose masks a part of the visual field, a camera's body blocks its own view. Image stitching attempts to compensate for this loss. Distortion can appear as misalignment, ghosting, and disappearance of objects as the views from a camera's front and back lenses overlap at the stitching points.

Figure 3 shows one frame in equirectangular projection at the same time stamp for the four cameras—all display stitching distortion. The GoPro shows misalignment and disappearance but the least amount of ghosting. The Insta360 shows all three kinds of distortion. The Garmin has some misalignment and ghosting and the greatest amount of disappearance. The Ricoh also features a sizable level of all three distortions and is worse than the Insta360.

4.2 Camera response to varying light intensity

To get a sense of how the cameras handle different light conditions, we used a dimmer to gradually reduce the light level in the laboratory. The light level was adjusted and

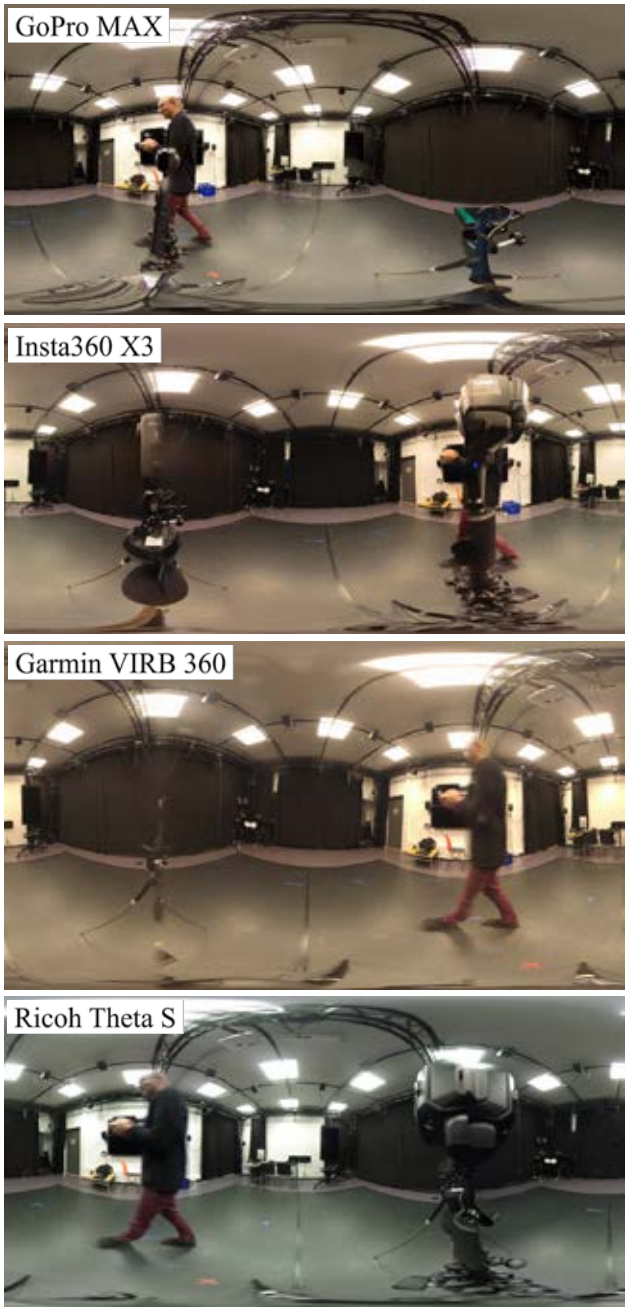


Figure 3. Equirectangular projections of the four camera recordings in the same frames, showing the different ways of stitching distortions at different positions.

held for a few seconds at each intensity, resulting in eight light intensity levels. A Samsung Galaxy S23 Ultra mobile phone was set to record the light intensity in the room with the Android app Physics Toolbox Sensor Suite 1, developed by Vieyra Software (ver. 2023.01.21). The RGB and Brightness graphs for each camera are plotted in Figure 4. The graphs demonstrate the differences in the cameras as they adjust to eight lighting levels. The RGB adjustments manifest in different levels of red, blue, and green in the graphs in any lighting condition. The plot shows blue as the highest intensity, followed by green and red. The Ricoh only does a little RGB color adjustment with bright light and virtually nothing when the light intensity decreases.

Cameras manipulate RGB levels to allow for automatic white balance adjustment as the lighting changes. This ensures that the white parts of the captured frame appear neutral regardless of the lighting level, which helps maintain an accurate representation of colors. Some cameras have an additional automatic color enhancement feature to produce aesthetically pleasing, vibrant videos. Besides automatic white balance and color enhancement, video recordings often undergo exposure compensation [7]. This feature, again through RGB adjustment, produces an optimal brightness level by preventing over- or under-exposure in sub-optimal lighting conditions. The solid black line in Figure 4 shows the brightness level, essentially a function of the RGB values first converted to grayscale. All four cameras respond to all lighting levels. As a general trend, the RGB and brightness values dip first and then increase to a new steady state.

4.3 Projection and compression distortion

While all four cameras use a dual-fisheye lens setup, the manufacturers designed different algorithms to save their files as described in Chapter 3. The projection choices and the compression algorithms that work on the selected projection will create different artifacts. Figure 1 shows a frame of the original files from the cameras.

The GoPro MAX uses a modified equi-angular cubemap, making two 4096 x 1344 streams of video, each compressed to a bitrate of about 30,000 kbps. However, since the two streams have different fields of view (FOV), the equivalent per-angle bitrate of the two streams is different.

The Insta360 X3 saves the two fisheye lenses in separate files, with a rectangular resolution of 2880 x 2880 each. This makes it the highest equivalent full-360 resolution among the four cameras. Combined with the fps of 29.97, it also has the highest equivalent full-360 bitrate.

The Garmin VIRB 360 uses a standard equirectangular projection, with the second-highest bitrate of 80,000 kbps. However, due to the polar expansion of the equirectangular projection, some redundant pixels near the poles are encoded with more than enough bits. Compared to the EAC projection without poles or the dual-fisheye with easily compressible black edges, it is a less compact method for file storage.

The Ricoh Theta S saves the two fisheye lenses into a single video file. Not only does it have the lowest bitrate, but it also saves the file as a standard 1080p 16:9 rectangular video, which is not in the 2:1 ratio of the dual-fisheye image. This resulted in the extra black edge at the bottom and the abnormal 1920x960 resolution in the software-exported equirectangular video, as shown in Figure 3.

5. EXTENDING THE MUSICAL GESTURES TOOLBOX

The overview above shows that all cameras record video with different projections, file formats, pixel dimensions, and frame rates. This makes it challenging to establish uniform workflows when handling recordings from different cameras. We have, therefore, implemented tools in our

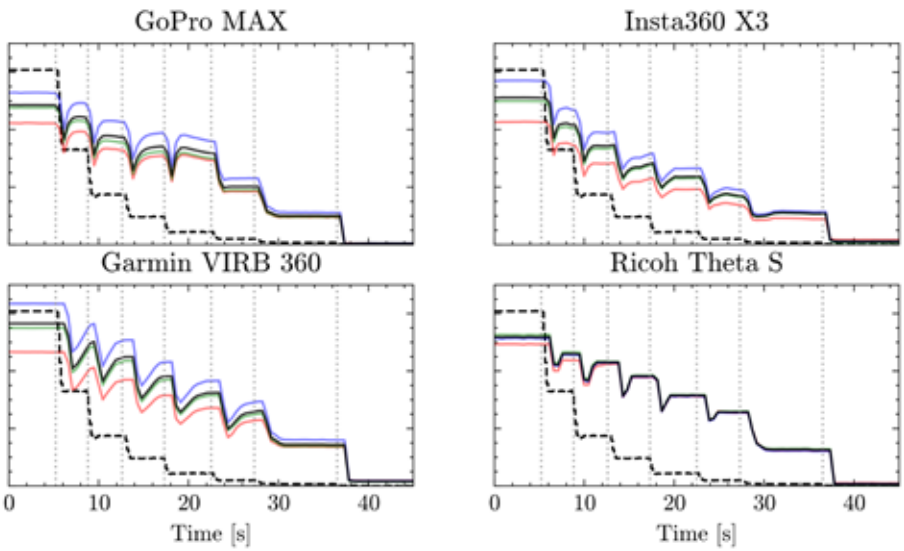


Figure 4. RGB (colored) and summed (solid black) light intensity of the four cameras compared to light intensity captured by the phone (dashed line). The units are different: the phone captured the intensity in lux, while the cameras' intensity value is the averaged pixel value ranging from 0 to 255. The cameras are calibrated by their internal algorithms.

Musical Gestures Toolbox (MGT) [8] for handling the different file types so that they can be used for further analysis. This is implemented in a new class, `Mg360Video`, that contains utility functions for 360-degree video merging, projection conversion, etc. The class inherits from the main video class `MgVideo`, meaning all the current demonstration and analysis functions in the toolbox can now be used directly on 360-degree videos.

5.1 File merging

Both the Insta360 camera and GoPro MAX cameras store high-quality video in multiple files using proprietary formats (.INSV and .360, respectively). These segments needed to be concatenated into one file to facilitate subsequent analysis. Fortunately, both the .INSV and .360 files can be read by FFmpeg. Since they contain video streams that use standard video compression inside (H.264 and H.265, respectively), storing the streams in a standard container is possible. In the updated `MgVideo` and `Mg360Video` classes, we now support inputting a list of filenames that will automatically be merged using FFmpeg's `concat` function. We have decided to export files in the Matroska file format (.MKV), an open format that supports unlimited video, audio, picture, or subtitle tracks in one file. This is a versatile video container widely compatible with various software applications. This approach ensures that the exported content maintains the original video and audio specifications with minimal alteration.

5.2 Projection conversion

The `Mg360Video` class supports projection conversion with the `v360` filter function of FFmpeg [7]. This filter supports conversion between several common 360-degree video formats. When creating a new object, the original projection information is needed. Then, it can be converted into more than twenty different projections using the `Mg360Video.convert_projection()` function, as shown in the first row of Figure 5.

5.3 Motion analysis

The Musical Gestures Toolbox supports various motion analysis functions that are now compatible with 360-degree videos, including motion average/history images/videos, videograms, motiongrams, self-similarity video matrices, etc. The second row of Figure 5 demonstrates a motiongram in the vertical direction and an extracted motion video frame.

6. CONCLUSIONS

Through desktop investigations and a lab experiment, we have found that the four 360-degree cameras in question all have different video qualities, file formats, projections, and software support. Given their (relative) commercial success, we are surprised by the lack of technical documentation and file conversion support. The GoPro MAX has the best overall video quality, good software, and detailed metadata. Still, its proprietary .360 file format with a customized video projection makes it hard to work with the files from a video analysis perspective. The Insta360 X3 has the highest original resolution, good software support, and unique stitching adjustment functionality. The fish-eye files can be merged and stitched using the Musical Gestures Toolbox, thus making it easier to do batch processing and analysis. Given their lower visual quality, we would not recommend the two legacy cameras (Garmin VIRB 360 and the Ricoh Theta S) for research purposes. However, since we (and others) have numerous recordings from these cameras in our archives, we must also find solutions for handling these files.

[7] `https://ffmpeg.org/ffmpeg-filters.html#v360`

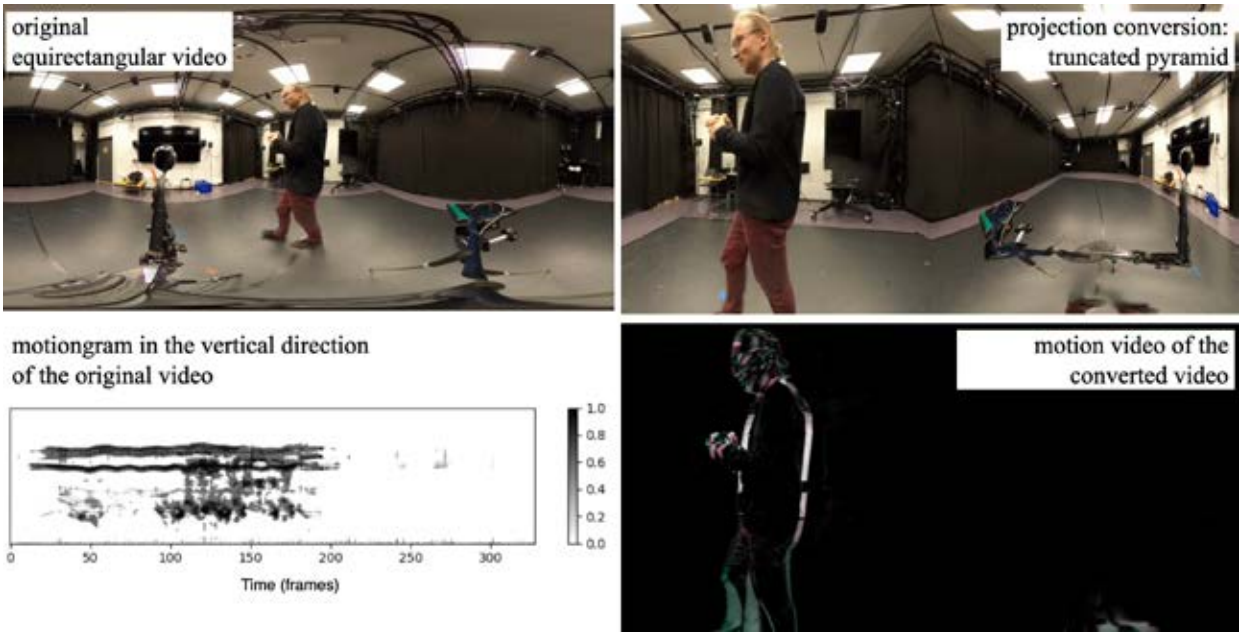


Figure 5. Examples of the new functionality of the extended Musical Gestures Toolbox: projection conversion, motiongram analysis, and motion video generation.

Support for 360-degree videos in the Musical Gestures Toolbox for Python allows researchers to perform file merging and simple projection conversions in a package that already supports numerous video visualization and analysis pipelines. This is crucial for our sonic environment research and will hopefully inspire others to add 360-degree recordings to their data collection.

Moving forward, we hope that manufacturers can work together to support a standardized, open-source representation of 360-degree videos that contain necessary metadata, including projection and stitching information. This can hopefully be the starting point for an open software ecology that supports recording, processing, editing, and analysis from all 360-degree cameras. We will also keep updating the 360-degree video-related functionalities in the Musical Gestures Toolbox, such as 360 video cropping and support for raw camera output formats.

Acknowledgments

This project is supported by the Research Council of Norway through projects 262762 (RITMO), 250698 (MICRO), 324003 (AMBIENT), and 322364 (fourMs).

7. REFERENCES

[1] Netflix Production Technology Resources, "Into the Volume: A Behind-the-Scenes Look into the Virtual Production of 1899," Dec. 2022. [Online]. Available: https://www.youtube.com/watch?v=ZMynJCgJIQk

[2] F. Chiariotti, "A survey on 360-degree video: Coding, quality of experience and streaming," *Computer Communications*, vol. 177, pp. 133–155, Sep. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S014036642100253X

[3] R. G. d. A. Azevedo, N. Birkbeck, F. De Simone, I. Janatra, B. Adsumilli, and P. Frossard, "Visual Distortions in 360° Videos," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2524–2537, Aug. 2020, conference Name: IEEE Transactions on Circuits and Systems for Video Technology. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8756213

[4] D. Salomon, *Transformations and projections in computer graphics*. London: Springer, 2006, oCLC: 209949994.

[5] M. Riaz, J. Guo, and A. R. Jensenius, "Comparing Spatial Audio Recordings from Commercially Available 360-degree Video Cameras," submitted.

[6] F. Cutolo, N. Cattari, U. Fontana, and V. Ferrari, "Optical see-through head-mounted displays with short focal distance: Conditions for mitigating parallax-related registration error," *Frontiers in Robotics and AI*, vol. 7, p. 572001, 2020, publisher: Frontiers Media SA.

[7] S. Bianco, A. R. Bruna, F. Naccari, and R. Schettini, "Color correction pipeline optimization for digital cameras," *Journal of Electronic Imaging*, vol. 22, no. 2, pp. 023 014–023 014, 2013, publisher: Society of Photo-Optical Instrumentation Engineers.

[8] B. Laczkó and A. Jensenius, "Reflections on the Development of the Musical Gestures Toolbox for Python," in *Proceedings of the Nordic Sound and Music Computing Conference*, Copenhagen, 2021. [Online]. Available: https://www.duo.uio.no/handle/10852/89331

EXPLORING CONVOLUTIONAL NEURAL NETWORK MODELS FOR MULTIMODAL CLASSIFICATION OF EXPRESSIVE PIANO PERFORMANCE

Anna-Maria CHRISTODOULOU[1,3], **Sagar DUTTA**[1,3], **Olivier LARTILLOT**[1,3], **Kyrre GLETTE**[2,3], and **Alexander Refsum JENSENIUS**[1,3]

[1]*Department of Musicology*, **University of Oslo**, Oslo, Norway
[2]*Department of Informatics*, **University of Oslo**, Oslo, Norway
[3]*RITMO, Center of Excellence for Rhythm, Time, and Motion*, **University of Oslo**, Oslo, Norway

ABSTRACT

This paper addresses improving performance analysis by automating the recognition of expressive performance styles. We propose a multimodal fusion approach integrating audio, video, and motion data. We demonstrate the effectiveness of our approach by utilizing convolutional neural network (CNN) models. Training is done on a classical piano dataset of 211 excerpts containing audio, video, MIDI, and motion capture data. The results highlight the robustness of the CNN models; they achieve high accuracy even when trained on a limited dataset. Our study contributes to advancing the field of performance analysis by applying deep learning techniques to multimodal data.

1. INTRODUCTION

Musicians use many types of body motion while performing, ranging from sound-producing actions to expressive gestures that may not directly affect the musical sound [1]. Here, we are interested in music performance styles, especially those related to piano performance. Music performances are multimodal in nature. We define multimodality as the combination of diverse data types that offer complementary information to the processing task, typically arising from perceived relationships between sound-producing actions and the resulting sounds. However, they may also be evoked from only auditory or visual modalities. Musical actions vary and can help shape the musical phrasing, express emotions, or communicate a structural change in the piece. Interestingly, even subtle actions can be perceived as significant and influence the conveyance of expressive elements; moving an eyebrow or gesturing with a finger can be powerful artistic expressions.

Humans can easily observe and understand the many layers of musical actions in performance, and professional musicians, educators, and music researchers have a highly developed sense of such bodily nuances. However, many challenges arise when training machine learning systems to identify musical actions in a dataset. Convolutional Neural Networks (CNNs), a type of deep learning model particularly effective in processing visual and spatiotemporal data, have shown great promise in this area. CNNs can automatically learn and extract features from input data, making them suitable for recognizing complex music performance patterns. Such an *automated expressive music performance recognition* task is central when analyzing music performances.

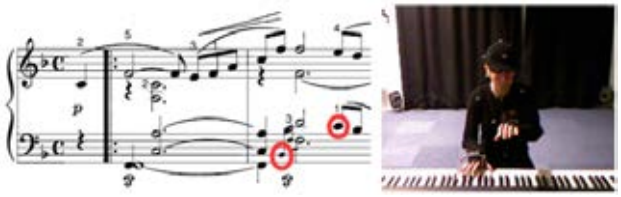


Figure 1. A left-hand phrase from Traumerei (left), between the circled notes, and a still image from video (right) showing normal tempo, exaggerated expression. [2]

Our research is driven by the goal of exploring the potential of integrating multimodal approaches and deep learning models in automated expressive music performance recognition during music performances (Fig. 2). We specifically address the challenges of combining audio, video, and motion information while identifying unexplored possibilities for advancing the recognition and analysis of expressive music performance. We aim to answer the following research questions:

- Can a 1D CNN accurately detect spatiotemporal information from piano performances?
- How can we work with multimodal CNNs for a music performance classification task, and how many dimensions are necessary?
- Is there an effect in accuracy between unimodal and multimodal approaches?

In the following sections, we analyze related work and discuss the effectiveness of multimodal approaches in music recognition tasks. We provide details about the dataset used, as well as the architecture of our model, and share our observations on the obtained results. Through this initial exploration, we seek to establish the groundwork for future strides in multimodal music performance recognition. [1]

[1] Our pre-trained models and our code are publically available with an

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

Music experiences are inherently multimodal. In a concert, we *see* musicians playing on their instruments, *hear* the resultant musical sound, *smell* people around us and the environment, *feel* the vestibular motion when moving or dancing to music or the sound waves reaching our bodies in a loud concert [3]. Music may even change the *taste* of the food we are eating [4]. Similarly, when performing, we *touch* the instrument or *move* our body to the music, we *hear* the sound and *see* the sheet, the instrument, or the conductor instructing us how to play [5]. Thus, to understand musical experiences, it is necessary to employ analysis techniques that allow for studying the complexity at hand in a complex and multimodal manner. However, significant gaps remain, particularly in integrating multiple modalities to capture and characterize musical expressive performances comprehensively.

Past research has shed light on how individuals perceive and represent musical phrases through bodily movements [6]. Understanding humans' mapping strategies provides insights into how we interpret and express music through body movements. Such findings have implications in various domains, such as music performance, education, and human-computer interaction, adding to our understanding of individual and cultural variations in gestural interpretations of music. This can also contribute to our understanding of the embodiment of music and show the importance and relevance of research in music expression analysis, as well as related fields.

Sarasua showed that multimodal music analysis outperforms unimodal approaches in various music processing tasks, including music gesture recognition [7]. They explored context-aware gesture recognition of a conductor's gestures, incorporating audio and motion capture data. This involved clustering similar gestures based on extracted motion capture and audio features. The study showcased the potential of analyzing and interpreting the conductor's expressivity during musical performance. However, video data, which could provide vital visual cues, was not included.

Wu et al. [8] proposed a classifier combination framework for action recognition using audio and motion capture. Their approach involved extracting audio features, such as Mel-frequency cepstral coefficients (MFCCs), and motion features related to joint angles and accelerations. By combining these features, they achieved an accuracy improvement of 6% compared to previous single-modal approaches.

Chang et al. [9] focused on modernizing music recording using classification techniques. Their automated analysis pipeline for piano playing sessions utilized multiple data types, including text, audio, video, and electroencephalography (EEG). Their approach involved extracting features such as pitch salience, dynamics, and spectral flux from the audio signals and capturing video data of hand motion.

Outside music-related research, Nandakumar et al. [10] presented a novel multimodal action recognition approach based on skeletal joint positions. Their approach included a spotting module to detect gesture instances and a recognition module to classify the gesture types. They achieved an average recognition accuracy of 87%.

In summary, while previous studies have significantly contributed to action recognition, gaps remain, particularly in integrating audio, video, and motion data to characterize expressive music performances. By incorporating multiple modalities in the dataset and employing advanced machine-learning techniques, we aim to open new musical interpretations and gain a deeper understanding of the relationships between music sound and motion.

3. METHODS

This paper used a multimodal music dataset to study expressive piano performances. The dataset included audio, video, and motion capture. After a thorough data preprocessing, we implemented three classification rounds, one for each data type. We then concatenated the separate representations into a Multimodal CNN classifier. Throughout our experiment, we also integrated regularization techniques to improve generalization and convergence.

3.1 Dataset Description

For this paper, we used a multimodal piano dataset from Sarasua et al. [2] to analyze expressive music performance. The dataset is available under the CC BY-NC-SA license [11] and comprises audio, video, motion capture, and MIDI data. The participating pianists performed an excerpt of Robert Schumann's Träumerei (Kinderszenen Op.15 No.7) (Fig. 3).

The dataset is of two pianists (*pianist01* and *pianist02*) who perform the excerpt in three conditions (normal, slow, and fast), with and without a metronome. Each condition was recorded three times. One exception was that pianist01 had two recordings for the conditions metronome, exaggerated performance, and normal speed. Thus, the complete dataset contains 211 samples for the two pianists. In each of these conditions, their expressive performance can be categorized under the following five expressive styles:

- **Normal:** a standard performance without specific emphasis or modifications.
- **Still:** the pianists aimed to minimize their actions, keeping their bodies and hands as still as possible while playing, while the resulting sound was consistent with minimal variation.
- **Exaggerated:** exaggerated actions and expressions, where the pianists intentionally emphasize certain changes in dynamics and articulation.
- **Finger Legato:** smooth hand actions and connected sounds by emphasizing finger legato technique without noticeable breaks or pauses.
- **Finger Staccato:** short and distinct sounds using a finger staccato technique and sharp, quick hand actions.

open license: DOI10.17605/OSF.IO/Y72CG

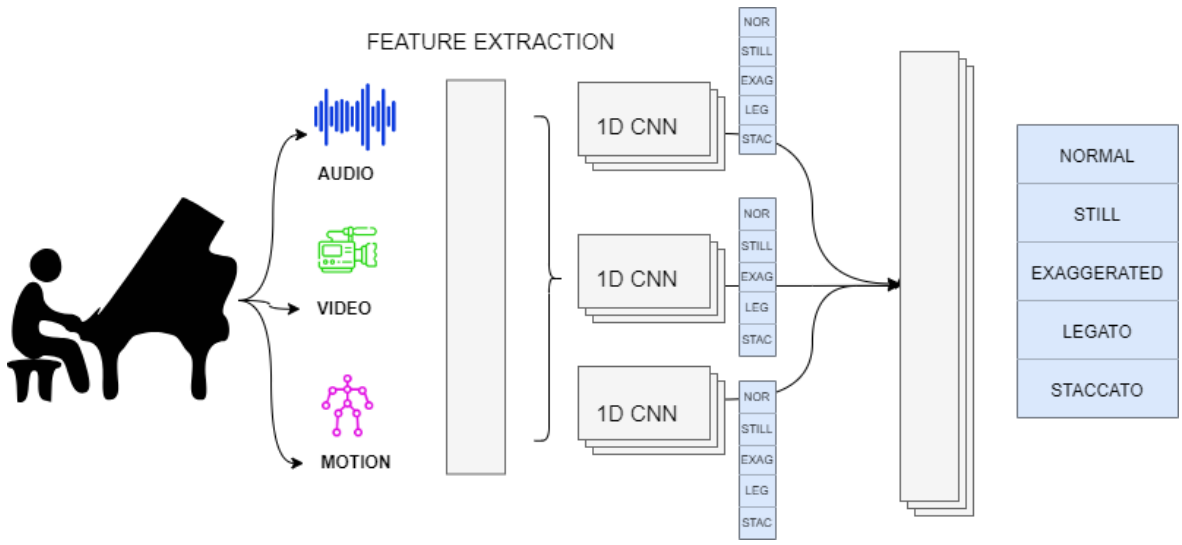


Figure 2. An overview of the Multimodal Fusion Method. The model receives three types of input data from the pianist's recordings (features extracted from audio, video, and motion capture). These inputs are processed and fed into a 1-dimensional CNN, each providing a classification result. The representation layers of the individual CNNs are then fed into the Multimodal CNN for the final classification output.

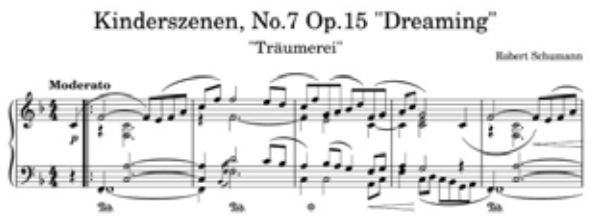


Figure 3. The excerpt from Kinderszenen Op.15 No.7 from Robert Schumann's Träumerei used in the dataset recordings we analyzed.

The audio was recorded at a sampling rate of 44.1 kHz, the video at 30 frames per second (fps), and motion capture data at 100 Hz. The motion capture data includes 22 features (position of body parts), but only two are retained for this project: the right and left hand in three dimensions. The recorded performances range from 8 to 48 seconds, the latter being the slow-speed recording. This provided a diverse data set for analysis across modalities.

3.2 Feature Extraction

Features were extracted from audio (RMS, spectral contrast), video (motiongrams), and motion capture data (velocity, acceleration). All of the features were padded to the maximum length of the input data (different for each data type). Padding is essential in data processing for machine learning models, particularly when dealing with sequences of different lengths. Consistent input shapes are maintained by padding data to a maximum length. This enables seamless integration with models that expect fixed-size *tensors*, algebraic objects that describe a multilinear relationship between sets of algebraic objects related to a vector space. The features were also concatenated with the corresponding labels attached to each sample and implemented one-hot label encoding. This means the labels for our five different expressive categories were transformed from numerical values between 0 and 4 to a sequence of 0 seconds. By implementing one-hot label encoding and concatenating the labels to each sample, we ensure that the labels are adequately represented for model compatibility, loss calculations, evaluation metrics, and ease of integration. This enables effective training and evaluation of machine learning models in our classification task.

3.2.1 Audio

Two features were extracted from the audio, using Librosa [12] in Python: the amplitude level and the spectral contrast [13]. The average amplitude level, calculated as the root mean square (RMS), reflects the audio signal's overall energy content and indicates the dynamic nature of the performance. Variations in touch, articulation, and dynamics manifest as distinct RMS values, allowing for differentiation between soft and gentle or forceful and intense actions. The spectral contrast captures disparities in spectral energy across frequency bands. This feature discerns the unique spectral profiles of diverse playing techniques, such as staccato and legato, marking the significance of frequency-specific variations in expressive piano playing.

3.2.2 Video

We extracted the vertical and horizontal motiongrams from the video data. Motiongrams represent the intensity of motion in a video. They are obtained by calculating the pixel differences between consecutive frames, both horizontally and vertically [3]. Let M be a motiongram array of shape $(w+1, n)$, where w represents the width of the motiongram and n is the total number of frames. We can compute the 1D feature vector F of length $w+1$ as follows:

$$F_j = \frac{1}{n}\sum_{i=1}^{n} M_{ji}, \quad j = 1, 2, \ldots, w+1$$

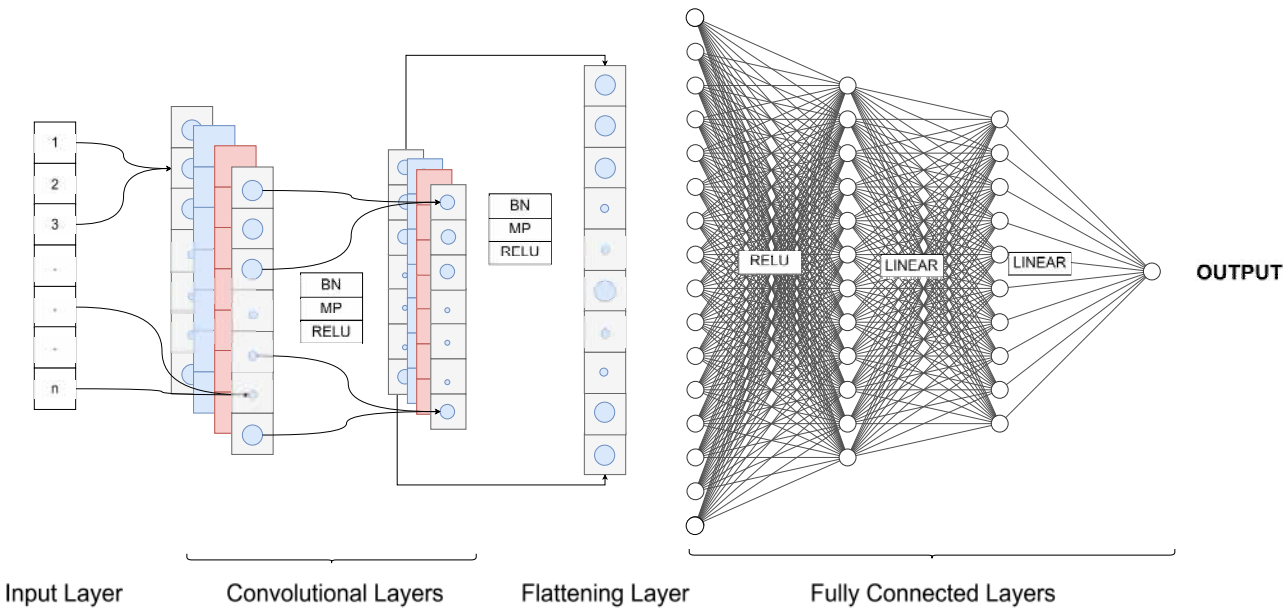


Figure 4. Illustration of the 1D Convolutional Neural Network (CNN) architecture designed for analyzing audio, video, and motion capture data. The architecture consists of three 1-dimensional convolutional layers (conv1, conv2, conv3) with 16, 32, and 64 output channels, respectively, followed by batch normalization and max pooling. The flattened output is then passed through fully connected layers (fc1, fc2, fc3, fc4, fc5) with the final softmax output layer supporting classification into different classes based on the unique characteristics of the input data.

The equation represents the vertical averaging process, where F_j is the j-th element of the 1D feature vector F and M_{ji} represents the motion value in the j-th column and i-th frame of the motiongram array M. These computed features F capture the overall motion patterns in the video and can be used for various analysis or modeling tasks.

We averaged the horizontal and vertical motiongrams in one dimension. This resulted in a concise visualization of the motion intensity throughout the video, helping us compress our motion information while dimensionalizing it and preserving the relevant temporal patterns.

3.2.3 Motion Capture

The motion capture data represents the instantaneous position (in 3 axes) of the participants' right and left hands. We extracted velocity and acceleration from these, using NumPy's gradient function to compute the derivatives of the positional data. Each hand's resulting velocity and acceleration arrays were then stacked to form a feature matrix, which was saved as a NumPy binary file for further analysis and processing. This enabled comprehensive motion data to be extracted from the original motion capture files. To check for overfitting, we visualized the resultant vectors of the motion for the left hand and right hand.

3.3 Model Description

CNNs demonstrated remarkable performance in capturing spatial and temporal patterns within data, making them suitable for analyzing time-series data [14]. A 1-dimensional Convolutional Neural Network (1D CNN) was built using Python for each modality (audio, video, and motion capture) (Fig. 4). The architecture comprised three 1-dimensional convolutional layers (conv1, conv2, conv3) with 16, 32, and 64 output channels, respectively, each followed by batch normalization [15] and max pooling [16]. The convolutional layers had a kernel size of 2, a stride of 1, and padding of 1 to maintain the input dimensions. The output of the convolutions was flattened and passed through fully connected layers (fc1, fc2, fc3, fc4, fc5) with 256, 128, 64, 32, and 5 (according to the number of classes) output nodes, respectively. Batch normalization was applied after each fully connected layer. ReLU activation functions were used after each layer to introduce non-linearity. The consecutive linear layers were designed to progressively transform and refine the feature representations, capturing complex patterns and relationships that might not have been possible with a single linear layer. Each layer added a level of abstraction, allowing the model to learn more nuanced features.

The 1D CNN allowed the exploitation of the temporal nature of audio data. Using convolutional layers, the model could learn and detect temporal relationships between different sound segments, extracting important features for classification. The hierarchical representations obtained by the convolutional layers captured both local audio features and global audio characteristics, enabling the model to effectively distinguish between different audio classes.

Similarly, it enabled the network to capture temporal dynamics in video analysis. The model discerned motion information contributing to different classes' visual representation by convolving along the temporal dimension. Combining convolutional layers and batch normalization aided in learning robust visual features, while the fully connected layers captured complex relationships between these features.

A 1D CNN was also well-suited for processing the sequential nature of motion capture sequences. The convolutional layers learned spatial dependencies over time, capturing joint position and motion changes throughout the sequence. This enabled the model to extract discriminative motion features for classification.

Batch normalization after each convolutional layer stabilized the learning process by ensuring consistent feature scaling and reducing the impact of covariate shifts. The rectified linear units (ReLU) introduced non-linearities, allowing the model to learn complex representations.

The final softmax output layer was designed to support classifying inputs into different classes, potentially offering predictions based on audio, video, and motion capture data. It was the last activation function of a neural network, normalizing the network's output to a probability distribution over predicted output classes. By utilizing 1D CNNs for each modality, the model aimed to exploit the unique characteristics and temporal dependencies inherent in audio, video, and motion capture data, potentially yielding improvements in classification performance.

3.4 Training process

In the two-stage training process, the data is utilized as follows: in the first stage, the CNN is trained on the individual modality data (audio, video, or motion capture) independently to learn modality-specific characteristics and temporal dependencies. In the second stage, the pre-trained CNNs for each modality are combined and further trained on the multi-modal data to capture cross-modal correlations and enhance classification performance. This two-stage approach leverages the unique temporal dynamics of each modality and effectively integrates information from multiple data sources.

In more detail, we began training by developing distinct CNN models for individual modalities, including audio, video, and mocap data. We had 211 vectors for each modality, with lengths of 431, 1225, and 9794, respectively. Each CNN model is trained to autonomously extract relevant features and embeddings from its respective input data. Upon training completion, the trained CNN models are utilized to generate embedding vectors from the input data of their corresponding modalities by extracting the outputs from the convolutional layers of each CNN. These outputs are the learned representations or embeddings that capture the temporal and spatial patterns specific to the audio, video, and motion capture modalities. This approach allows leveraging the learned features and representations from the trained CNNs to extract valuable information from the input data across different modalities.

Subsequently, we perform k-fold cross-validation to evaluate the models' generalization and assess potential overfitting by training and validating the model on different subsets (folds) and aggregating the performance metrics to assess generalization. This process involves partitioning the dataset into k equally sized folds, iteratively training the models on $k-1$ folds, and evaluating their performance on the held-out fold. We used 5 folds. This enables us to obtain a more comprehensive understanding of the models' robustness and generalization capabilities.

3.5 Fusion Process

Following the generation of embedding vectors for each modality, we utilize the concatenation fusion technique to merge the vectors and create a unified representation of the input data. The fused embedding vectors are then fed into our CNN model, which consists of two 1D convolutional layers. The first convolutional layer has 16 output channels, a kernel size of 3, a stride of 1, and padding of 1. The second convolutional layer has 32 output channels, a kernel size of 3, a stride of 1, and padding of 1.

In addition to the convolutional layers, our model incorporates batch normalization after each convolutional layer to normalize the activations and enhance training stability. ReLU activation functions are applied after batch normalization to introduce non-linearity. Max pooling operations are performed with a kernel size of 2 and a stride of 2 to reduce the spatial dimensions of the feature maps. The output of the convolutions and pooling layers is then flattened and passed through a fully connected layer. The fully connected layer maps the flattened input to the number of output classes.

4. RESULTS

Our method showed that 1D CNNs can accurately learn spatiotemporal patterns and classify expressive piano performance styles. The unimodal CNN classification results showed that the video CNN achieved an accuracy of 70.57%, the audio CNN attained an accuracy of 86.79%, and the motion CNN achieved an accuracy of 67.92%. In contrast, the multimodal classification model, which fused the information from the video, audio, and motion modalities, achieved an accuracy of 95.29%.

Future work would include comparing single and multiple dimensions in the CNN architecture to see how they affect the classification and learning of the CNN models. Working with other types of architectures, such as Long Short Time Memory (LSTMs) or transformers, would also be interesting, since they effectively handle sequential data, capturing detailed temporal dynamics and complex relationships in music. These models leverage attention mechanisms and long-range dependencies, enabling robust and scalable performance in discerning subtle expressive elements in musical performances. Another interesting approach would be to test audio–video, audio–mocap, or video–mocap combinations and compare their effects on classification performance.

5. DISCUSSION

This study explored multimodal data integration for expressive music performance classification. We incorporated three modalities: video, audio, and motion capture. To comprehensively understand their contributions, we trained separate Convolutional Neural Network (CNN) models for each modality and evaluated their classification performances. The results of our study revealed that the

multimodal approach outperformed individual CNN models. We achieved improved discriminative power and captured diverse aspects of the performances that were not discernible by analyzing each modality independently.

While our study demonstrated the benefits of multimodal data integration for expressive piano performance classification, some limitations should be acknowledged. First, acquiring multimodal data can be challenging due to the scarcity of synchronized multimodal recordings. Collecting a large and diverse dataset with annotated actions across multiple modalities requires significant resources and expertise.

Combining different modalities for fusion requires careful consideration of their compatibility. Not all modalities may effectively complement each other or offer significant synergies. Exploring the ideal combinations of modalities and their fusion techniques is an avenue for further research. Our study utilized a late fusion technique (after feature extraction), but various other fusion methods are available, such as early fusion, hybrid fusion, or attention-based fusion. Exploring different fusion approaches and optimizing the architecture design can improve classification performance.

Furthermore, the effectiveness of the multimodal fusion approach may depend on the specific set of actions and users included in the training data. The ability to generalize to new performance styles and users, especially in real-world scenarios, remains a challenge that warrants further investigation. Lastly, the increased complexity and fusion of multiple modalities may compromise the classification model's interpretability and explainability. Understanding and interpreting how the model arrives at its classification decisions becomes more challenging as the number of fused modalities increases.

6. CONCLUSIONS

Our research aims to lay the foundations for exploring the effectiveness of integrating multimodal approaches and deep learning models in automated expressive music performance recognition during music performances. Through our initial experiments, we have sought to uncover the potential of combining audio, video, and motion data to advance the analysis of expressive music performance.

With a small dataset, we achieved an accuracy of 95%, showcasing our approach's possibilities. By assessing the integration of audio, video, and motion data within a multimodal fusion framework, in conjunction with simple and computationally efficient deep learning models, we have begun to unlock the potential for enhancing the precision and efficacy of expressive music performance recognition. These initial findings provide valuable insights and set the stage for further exploration and refinement.

Moving forward, this research has the potential to inform pedagogy, personalize feedback based on individual abilities, enhance participation and enjoyment in music-making activities, and empower individuals from diverse musical backgrounds. This may open avenues for understanding and analyzing performance nuances, shedding light on a performer's technique, style, and emotional expression. Expressive music performance analysis opens for innovative applications, such as performance feedback systems, personalized music education platforms, as well as immersive and collaborative music experience tools.

Acknowledgments

This project is supported by the Research Council of Norway through project 262762 (RITMO).

7. REFERENCES

[1] A. R. Jensenius, M. M. Wanderley, R. I. Godøy, and M. Leman, "Chapter 2. Musical gestures: concepts and methods in research," 2010.

[2] Sarasúa, B. Caramiaux, A. Tanaka, and M. Ortiz, "Datasets for the Analysis of Expressive Musical Gestures," in *Proceedings of the 4th International Conference on Movement Computing*. London United Kingdom: ACM, Jun. 2017, pp. 1–4. [Online]. Available: https://dl.acm.org/doi/10.1145/3077981.3078032

[3] A. Jensenius, "ACTION – SOUND - Developing Methods and Tools to Study Music-Related Body Movement," Ph.D. dissertation, Jan. 2007.

[4] J. Thompson-Bell, A. Martin, and C. Hobkinson, "'Unusual ingredients': Developing a cross-domain model for multisensory artistic practice linking food and music," *International Journal of Food Design*, vol. 6, no. 2, pp. 233–261, Oct. 2021. [Online]. Available: https://intellectdiscover.com/content/journals/10.1386/ijfd_00032_1

[5] A. R. Jensenius, *Sound actions: conceptualizing musical instruments*. Cambridge: The MIT Press, 2022.

[6] T. Kelkar and A. Jensenius, "Analyzing Free-Hand Sound-Tracings of Melodic Phrases," *Applied Sciences*, vol. 8, Jan. 2018.

[7] A. Sarasua, "Context-aware gesture recognition in classical music conducting," in *Proceedings of the 21st ACM international conference on Multimedia*, ser. MM '13. New York, NY, USA: Association for Computing Machinery, Oct. 2013, pp. 1059–1062. [Online]. Available: https://dl.acm.org/doi/10.1145/2502081.2502216

[8] J. Wu, J. Cheng, C. Zhao, and H. Lu, "Fusing multimodal features for gesture recognition," in *Proceedings of the 15th ACM on International conference on multimodal interaction*, ser. ICMI '13. New York, NY, USA: Association for Computing Machinery, Dec. 2013, pp. 453–460. [Online]. Available: https://dl.acm.org/doi/10.1145/2522848.2532589

[9] X. Chang and L. Peng, "Intelligent Analysis and Classification of Piano Music Gestures with Multimodal Recordings," *Computational Intelligence and Neuroscience*, vol. 2022, p. e8232819, Jun.

2022, publisher: Hindawi. [Online]. Available: https://www.hindawi.com/journals/cin/2022/8232819/

[10] K. Nandakumar, K. W. Wan, S. M. A. Chan, W. Z. T. Ng, J. G. Wang, and W. Y. Yau, "A multi-modal gesture recognition system using audio, video, and skeletal joint data," in *Proceedings of the 15th ACM on International conference on multimodal interaction*, ser. ICMI '13. New York, NY, USA: Association for Computing Machinery, Dec. 2013, pp. 475–482. [Online]. Available: https://doi.org/10.1145/2522848.2532593

[11] "CC BY-NC-SA 4.0 Deed | Attribution-NonCommercial-ShareAlike 4.0 International | Creative Commons." [Online]. Available: https://creativecommons.org/licenses/by-nc-sa/4.0/

[12] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "librosa: Audio and Music Signal Analysis in Python," Austin, Texas, 2015, pp. 18–24. [Online]. Available: https://conference.scipy.org/proceedings/scipy2015/brian_mcfee.html

[13] Y. Jiang, C. Li, and Z. Wang, "Real-time Sound Visualization with Touch OSC: Stimulating Sensibility through rhythm in nature," 2019.

[14] M. Ashraf, F. Abid, I. U. Din, J. Rasheed, M. Yesiltepe, S. F. Yeo, and M. T. Ersoy, "A Hybrid CNN and RNN Variant Model for Music Classification," *Applied Sciences*, vol. 13, no. 3, p. 1476, Jan. 2023, number: 3 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/2076-3417/13/3/1476

[15] N. Bjorck, C. P. Gomes, B. Selman, and K. Q. Weinberger, "Understanding Batch Normalization," in *Advances in Neural Information Processing Systems*, vol. 31. Curran Associates, Inc., 2018. [Online]. Available: https://proceedings.neurips.cc/paper/2018/hash/36072923bfc3cf47745d704feb489480-Abstract.html

[16] H. Wu and X. Gu, "Max-Pooling Dropout for Regularization of Convolutional Neural Networks," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, S. Arik, T. Huang, W. K. Lai, and Q. Liu, Eds. Cham: Springer International Publishing, 2015, pp. 46–54.

COMBINING HAND POSE ESTIMATION WITH AUDIO FOR NOISE-ROBUST PIANO SCORE FOLLOWING

Kaitlin PET (kaitlinpet@gmail.com)[1] and **Akira MAEZAWA** (akira.maezawa@music.yamaha.com)[2]

[1]*Luddy School of Informatics, Computing, and Engineering*, **Indiana University Bloomington**, IN USA
[2]**Yamaha Corpration**, Hamamatsu, Japan

ABSTRACT

Tracking the musical score position from a user's piano performance is difficult in noisy environments and other cases when the audio signal is unreliable. We present a multi-modal score following method that seamlessly fuses audio and hand tracking from top-down video of a pianist's hands on the keyboard, using a probabilistic model and a feature-weighing mechanism. Experiments demonstrate our method's robustness under noisy environments and when pieces contain mistakes, compared to a score follower based only on audio input.

1. INTRODUCTION

Online score following allows a given input music performance to be continually temporally matched to its corresponding symbolic score [1, 2]. Score followers typically use performance audio as the input. They are often formulated as a Hidden Markov Model (HMM) whose states, emission probability, and transition probabilities are derived from the pitches and rhythms specified in the musical score [3–5].

Audio-based score followers become unreliable when the performance audio becomes unintelligible or inaudible, or the user plays unexpected notes that cannot be reasonably captured by the score follower's acoustic model. When audio cannot be relied on for score following, fusing complementary modalities can be helpful. For instance, during loud orchestral sections of a piano concerto, conductors might rely on visual cues from the pianist's body language. In a group music lesson where multiple students play the same piece on the keyboard in the same room simultaneously, a teacher might look at each student's hands and fingers to get an idea of what a student is playing. Audio-based score followers may also perform poorly if the pitches and rhythms being played differ from the underlying score, for example, in cases where the musician makes mistakes. When human ensemble members encounter a collaborator who makes mistakes, they may use a complementary source of information, like knowledge of common errors, to help interpret the performer's intention.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Previous research has shown that multi-modal score followers can mitigate tracking failures under low Signal-to-Noise Ratio (SNR) conditions like the situations described above. But, it is non-trivial to fuse the multiple streams seamlessly. One line of work incorporates additional inputs at a few carefully chosen locations [3, 6] such as before the beginning of a piece or after a *fermata*. Another line of work uses visual data as a complementary time series and chooses a media stream to track based on an estimate of reliability. For example, IMuSE automatic accompaniment system [7] uses multiple score followers operating on different modalities, including audio and the center of mass position of a pianist's left/right hands. During a performance, it uses tracking information from whichever follower has the smoothest series of inferred note onsets.

In general, the advent of deep learning-based pose frameworks such as MediaPipe [8] has made it easier than ever to accurately incorporate gestural information from a video feed into tasks in music informatics, such as multi-modal piano transcription [9–11]. Given such progress, it is increasingly important to explore its potential in score following and formulate a principled and simple mechanism for media fusion.

We present a multi-modal score follower that combines video and audio inputs for classical piano score following, leveraging hand and individual finger positions obtained from deep learning-based pose estimation alongside audio data. The fused score follower learns the expected audio and visual trajectory from a combination of the musical score and a "rehearsal" video of a pianist playing the music, then can track the pianist's hand and finger positions in a second, "performance" take. To integrate the audio and video streams seamlessly, we introduce a feature weighting mechanism to adjust the importance of each modality dynamically, enabling multi-modal real-time score following without the need for manual selection of modality. We show in two different environments that the fused score follower (1) can be tuned to largely retain the accuracy of an audio-only score follower when the audio signal is clear, (2) maintains a reasonable accuracy even when audio background noise levels are raised to the point that the piano playing is inaudible to the human ear, (3) outperforms an audio-only score follower when the pianist makes mistakes.

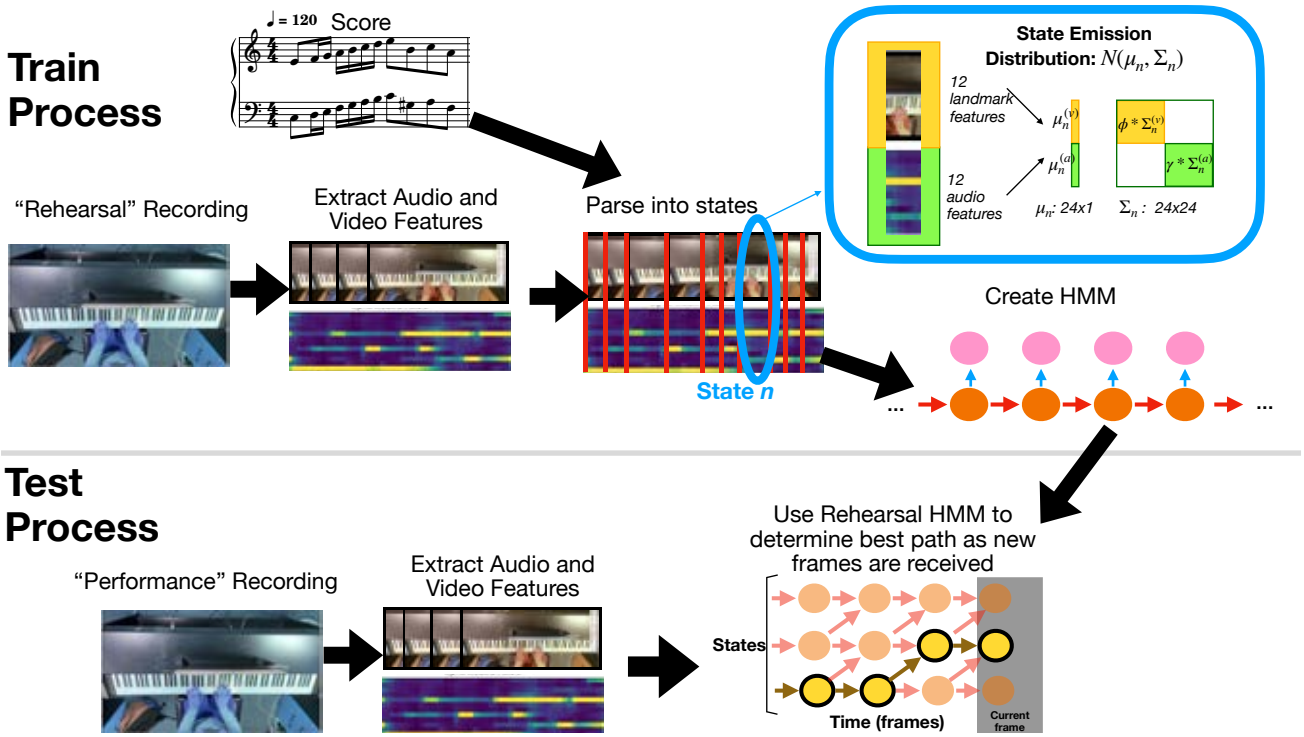


Figure 1. Overview of our method's training process (top), and test process (bottom).

2. METHODS

Our method matches a reference ("rehearsal") piano performance to a second performance of the same piece in real-time, as seen in Fig. 1. Tracking can rely on pitch-based features and hand landmark-based features. First, the "rehearsal take" of a piece is used to build the states of an HMM and set its emission and transition probabilities. To track the second "performance take" in an online manner, we use the rehearsal HMM to decode a state sequence, where state x_t in the decoded sequence is the state with the highest filtered probability at time t. Note that this method of online state assignment does not ensure a linear progression through the score – in the decoded state sequence, states are not guaranteed to appear in score-order and may jump around. This could make our state decoding method less-than-ideal for a real-world score following application. However, we find this method is sufficient for comparative evaluation of different score following strategies.

2.1 Hand landmark feature extraction

From the video stream, we extract a 12-dimensional position feature corresponding to the locations of 1) fingertips and 2) left hand/right hand center of mass, along the axis of the span of the keyboard. MediaPipe's Hand Landmarker [8] is first used to identify 21 "landmark locations" on each hand in every frame of the input video. Fig. 2 shows examples of labeled landmarks. In cases where a landmark is not detected, or a hand is absent from the frame, the missing landmark positions are set to their last observed positions.

The raw landmark positions are then adjusted using an affine transform to align with a pre-selected piano template. Before proceeding with further processing, alignment is visually confirmed for each recording. The transform is initialized based on the assumption that the middle finger of each hand would be positioned near the border between black and white keys. This initial transform is subsequently refined to maximize the enhanced correlation coefficient of the two images. This process ensures transformed coordinates are reliable and consistent for both takes, particularly along the horizontal axis, assuming that the middle fingers in the reference image were in the expected locations.

The Hand Landmarker model provides x, y, and z coordinates for each landmark point. Following the warping process, x denotes the horizontal position on the piano keyboard, y denotes the vertical position of the finger on the keyboard, and z denotes the finger depth with the wrist at $z = 0$, representing the extent to which the finger was raised or depressing a key. Preliminary analysis found that the trajectory of the x-coordinate exhibited the highest similarity across different takes of the same piece. This finding is logical since the horizontal position of the fingers directly corresponds to the notes being played. Although the trajectories of y coordinates showed moderate similarity, technical challenges during preprocessing led us to focus solely on the x coordinates for our landmark-based score follower.

Subsequently, we refine the landmark features by computing each hand's center-of-mass and retaining only the locations of the fingertips, yielding a total of 12 hand position features across both hands. These features were chosen because fingertips are directly involved in producing piano sounds, and hand center-of-mass was shown to be

Figure 2. Frames of top-down performance videos with MediaPipe landmarks colored. The top frame is from *no-mistake* dataset; the bottom frame is from *mistake* dataset

an effective tracking feature in Ritter et al. [7].

2.2 Multi-modal HMM for Score Following

The purpose of this study is to evaluate whether a fused audio-visual HMM-based score follower performs better than an audio-only HMM-based score follower. Thus, we opt for a simple, prescriptive approach in assigning states and transition probabilities. We focus more on the method to train the different emission probabilities, as these differ for visual and audio features.

We represent the score by a sequence of n hidden states, $\{x_1 \dots x_n\}$. These hidden states are matched to m frames of observed features $\{y_1 \dots y_m\}$. The features of y could consist of **(1) audio features** only, **(2) hand landmark features** only, or **(3) a combination** of the two.

States of the HMM are created by partitioning the piece's score by the union of note onsets and running sextuplets (1/6 beat units). For example, say a measure in the score has notes of length [2 beats, 1 beat, 1/12 beat, 1/12 beat]. This measure would translate to 20 total states, with 18 states corresponding to a 1/6 beat unit, and 2 states corresponding to a 1/12 beat unit.

Transitions from the current to the next state depend on the ratio of their score-based lengths. Formally, the probability of staying in state x_n is $\frac{l_n}{l_n+l_{n+1}}$, and the probability of moving from state x_n to state x_{n+1} is $\frac{l_{n+1}}{l_n+l_{n+1}}$, where l_n is the length (in beats) corresponding to the note value used to create state x_n. In the above example, l_n would correspond to 1/6 or 1/12. Backward transitions or skips have a zero probability.

Emission probability for each state x_n is a multivariate normal distribution constructed from 1) x_n's audio emission distribution, $N(\mu_n^{(a)}, \Sigma_n^{(a)})$, and 2) x_n's landmark emission distribution, $N(\mu_n^{(v)}, \Sigma_n^{(v)})$. We assume audio and landmark features are independent. Therefore, we can form μ_n by concatenating $\mu_n^{(a)}$ and $\mu_n^{(v)}$. We can form Σ_n by combining $\Sigma_n^{(a)}$ and $\Sigma_n^{(v)}$ in a blocked diagonal structure. To adjust the audio and visual features' relative contribution to the combined score follower, we introduce a video variance penalty parameter ϕ and an audio variance penalty parameter γ such that Σ_n is given as $\text{diag}([\gamma\Sigma_n^{(a)}, \phi\Sigma_n^{(v)}])$. See Figure 1 for a visual representation of this fusion method.

To create the audio-based emission distribution $N(\mu_n^{(a)}, \Sigma_n^{(a)})$, a "score reference audio" is first generated by rendering the symbolic score with the "Steinway Piano" virtual instrument in Logic Pro X [12]. The audio is transformed into a 12-feature constant-Q chromagram measuring the observed amount of each pitch class in the Western scale. Audio is then normalized such that each frame sums to one. MIDI from the rehearsal take is similarly translated to "rehearsal audio" in Logic Pro X, then transformed into a normalized chromagram. Next, the score reference chromagram and the rehearsal take chromagram must be parsed into states. Since our state definition is score-based, this means we first need a score match to the audio. In the score reference audio, state boundaries can be automatically computed by considering the tempo used to render the audio – since the score is translated literally, the "onset time" of each state is proportional to its position in the score. For the rehearsal audio, we first perform an offline score match using the *Orchestra* program [13]. Rehearsal audio may contain wrong notes, omitted notes, or additional notes (e.g. the player could play a short sequence of notes several times until it was error-free). These cases require manual correction to the alignment. If a note p is played at the wrong pitch, p's onset time is still set to the beginning of the intended note. If p is omitted, a note onset location for p is extrapolated based on the parsed note onset locations of notes $p+1$ and $p-1$. If a sequence of k notes corresponding to score positions $\{p...p+k\}$ is repeated, the note onsets times of $\{p...p+k\}$ are assigned to either the first or last repetition. After obtaining a score match, state onset times are determined by further segmenting the score via running sextuplets, as described above. For each state, $\mu_n^{(a)}$ is set to the sample mean of the state from the score reference chromagram – this way, expected pitch content would not be "polluted" by wrong, omitted, or repeated notes found in the rehearsal take. $\Sigma_n^{(a)}$ was computed by obtaining the sample covariance of frames in both the score reference chromagram and the rehearsal chromagram. Based on our state partitioning method, certain states may have insufficient unique frames to compute sample variance. We thus manually select a minimum covariance. This minimum covariance is set as the emission covariance should the state be too short for covariance computation; otherwise, this minimum covariance is added to the computed covariance values.

To create the landmark-based emission distribution $N(\mu_n^{(v)}, \Sigma_n^{(v)})$, we first obtain landmark trajectories for both hands' centers of mass and all fingertips using the pro-

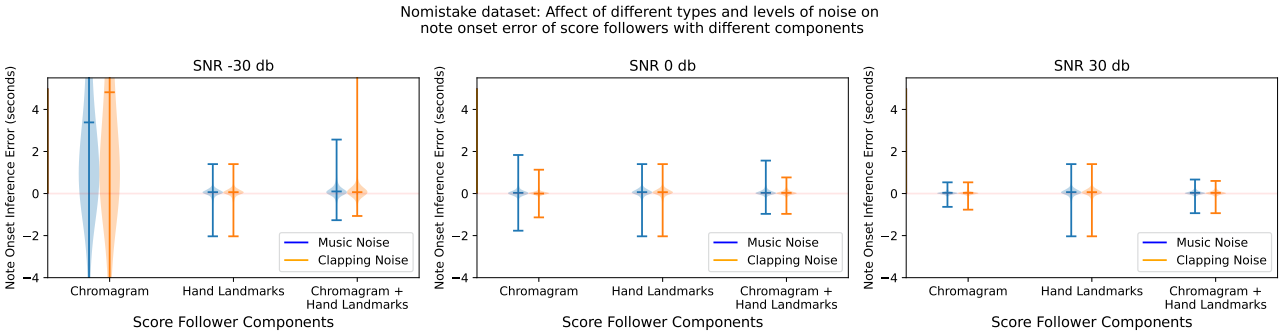


Figure 3. Violin plots showing the median, range and distribution of raw note onset error among three types of score follower (audio-only, landmark-only, and combined audio+landmark) on the *no-mistake* test set, given different noise types and SNR. Positive error values indicate the inferred onset time is late, negative error values indicate the inferred onset time is early. Violin plots show the full range of onset errors, as well as highlighting the median. Plots show behavior at chromagram weight $\gamma = 10$

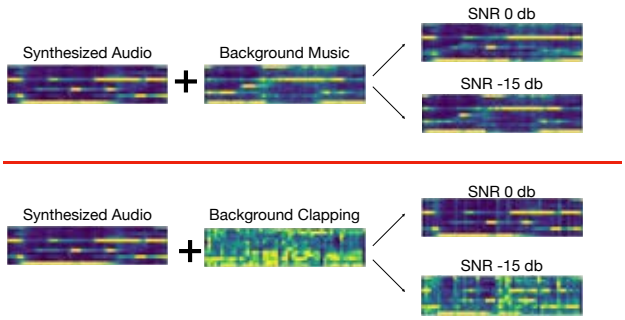


Figure 4. Two types of noise, music (a recording of Beethoven's *Symphony No. 5*) and clapping, are added in varying amounts to the synthesized piano. At higher SNR (*e.g.*, 0 dB), the original signal is still clearly visible, while at lower SNR (*e.g.*, -15 dB), the original signal is much harder to see and hear.

cedure described in Section 2.1. The trajectory sequences are then parsed into states using the state boundaries derived from the rehearsal audio. $\mu_n^{(v)}$ is set to the sample mean of landmark features in state n, and $\Sigma_n^{(v)}$ is based on sample covariance. In this computation, we assume three groups of independent features: the right hand's center of mass, the left hand's center of mass, and fingertips. A similar minimum covariance construct is used to ensure $\Sigma_n^{(v)}$ can is defined for all states.

3. EXPERIMENTS

We evaluated the effect of incorporating audio and visual features in our score follower, depending on the audio variance penalty γ. We also tested the performance of audio-only, landmark-only, and fused score followers when performances contained mistakes or varying amounts of background audio noise.

3.1 Data Collection

We obtained rehearsal and performance takes of multimodal piano performance data (two takes of each piece), capturing performance MIDI and top-down video. Data collection was performed with two pianists at different institutions (Yamaha and Indiana University Bloomington). See Figure 2 for sample frames from the two recording sessions. For the Yamaha dataset, Burgmüller's 25 etudes for piano were recorded by a late-intermediate to early-advanced pianist on a digital piano (Yamaha P125). The camera was suspended above the piano to give a top-down view covering most of the piano's width. As can be seen in Figure 2, a few of the leftmost keys were outside the camera's view. The player was not given specific performance instructions and was allowed to make mistakes. For example, when the pianist did not play a phrase to her satisfaction, she would sometimes repeat the phrase one or more times before continuing on with the performance. Of the 25 etudes, twelve were retained for analysis [1]. Six etudes were used for parameter tuning as described in Section 3.2, and six were used a validation set. **We will henceforth refer to the validation set from this recording session as the *mistake* dataset.**

The second dataset was collected at Indiana University Bloomington on a Roland electric piano (study done under the approval of Indiana University IRB #22017). This dataset was recorded to create an error-free version of pieces from the *mistake* dataset. In these recordings, the musician was explicitly asked to 1) not make mistakes and 2) play both rehearsal and performance takes with the same fingerings and hand distribution. We also positioned the camera in the second dataset to capture a top-down view. However, the camera in the second dataset was placed slightly higher so that all piano keys would be in-frame. **We will henceforth refer to data recorded in this session as the *no-mistake* dataset.**

For all performance takes, audio was synthesized from performance MIDI using the "Steinway Piano" virtual in-

[1] Two etudes contained too many mistakes, two contained pickup notes that were incompatible with the pipeline, and nine could not be effectively pre-processed for other reasons.

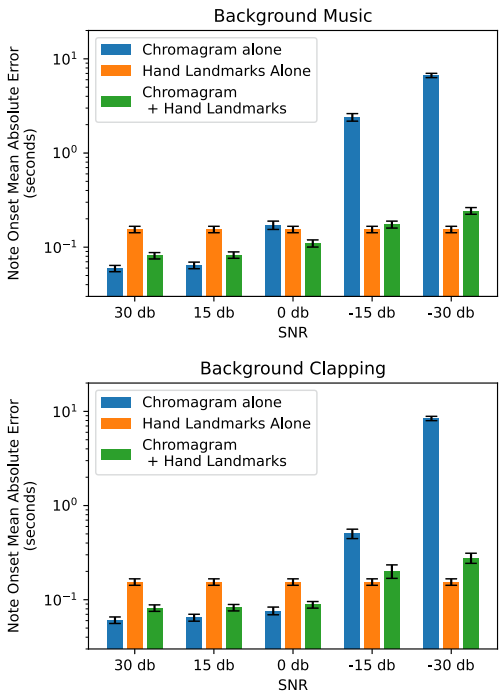


Figure 5. Highlighting behavior at the chromagram penalty level (γ) of 10. These graphs show the mean absolute note onset error of different score followers across a range of SNRs. Error bars represent +/- two standard errors.

strument in Logic Pro X [12]. To obtain ground truth onset labels, performance data (rendered audio and video recordings) were score-matched – *i.e.*, for each etude, all notes in its symbolic score were matched to a unique timestamp in the performance. This offline score match was achieved with a combination of custom code, the *Orchestra* program [13], and hand corrections, yielding a labeling resolution of 33 milliseconds. The *mistake* dataset lacked a one-to-one relationship between note onsets specified in the score and actual onsets in the performance, thus ground truth note onset labeling was performed using the onset labeling method described in Section 2.2.

3.2 Parameter Tuning

Six of the twelve etudes from the *mistake* dataset were used as a "tuning" dataset to hand-tune (1) ϕ, the penalty associated with the hand landmarks; (2) the minimum covariance for each data source; and (3) a good score-to-state partitioning strategy (See Section 2.2 for details). These parameters were used in creating the score following HMM trained by each rehearsal take. The same rehearsal-training pipeline described in Section 2.2 was used for both the *mistake* and *no-mistake* datasets.

Note that in the tuning set, the performer would sometimes play wrong notes, omit notes, or repeat a sequence of notes several times until she was ready to move on. These types of mistakes involve pressing keys with low horizontal distance to their correct counterparts (e.g. a wrong note can happen as the result of accidentally hitting the adjacent black or white key). Thus, picking $\phi > 1$ could allow the emission probability of a physically close mistaken note on a correct state to still be fairly high.

ϕ was chosen such that a score follower using only the landmark features worked reasonably well for every piece in a tuning dataset with mistakes. $\phi = 100$ was used in all experiments below, while γ, the audio variance penalty, varied depending on the experiment.

3.3 Experimental Conditions

Score followers were created to track three types of input data: (1) audio alone, (2) hand landmarks alone, and (3) combined audio and hand features. For each piece in the validation set, the rehearsal take was used to train three HMM-based score followers corresponding to these three types of input data. Each score follower was tested by feeding it audio and/or video of the performance take in 30-millisecond frames. After receiving each new frame, the score follower HMM was used to compute the state with the maximum probability in the current forward posterior distribution.

To evaluate the effectiveness of each score follower, we computed note onset error as the difference between the observed note onset times in the performance take and the inferred note onset times from the HMM. Given our online state determination method, certain states corresponding with note onsets may be skipped or repeated. We thus computed the observed time of each note onset state x_o as

$$\text{obs}(x_o) = \max\{\max(l(x < x_o)), \min(l(x \geq x_o))\}$$

where $l(x)$ is the frame in the test sequence where state x is observed. This definition ensures an onset time exists for every note. Below, we describe the effect on different performance conditions and score followers on onset error.

3.3.1 *Effect of Audio Variance Penalty*

As can be seen in Figure 6, our γ penalty was successful in weighing the behavior of the audio vs. video input streams in our fused score follower. We can see that at low γ values, the fused score follower more closely tracked the audio-only score follower's behavior, while at larger γ values, the combined score follower's behavior was more similar to the landmark-only one. Note that aside from weighing the audio and landmarks' relative strengths, γ also directly affected the emission variance of states in the audio-only score follower, and by extension, the audio-only score follower's note onset accuracy. When $\gamma < 1$, the audio-feature variance shrinks, potentially overfitting to the expected audio content and reducing accuracy when the performance differed from the rehearsal. If γ is increased too much, it can become difficult to distinguish between emission distributions of different score-based states. Notably, at the γ where audio score follower error is lowest, the fused score follower tends to also be at or near its lowest value. In Figure 6 we highlight a potential γ "sweet spot," $5 \leq \gamma \leq 20$, where the fused score follower's behavior is close to or better than whichever single-source score follower has higher accuracy.

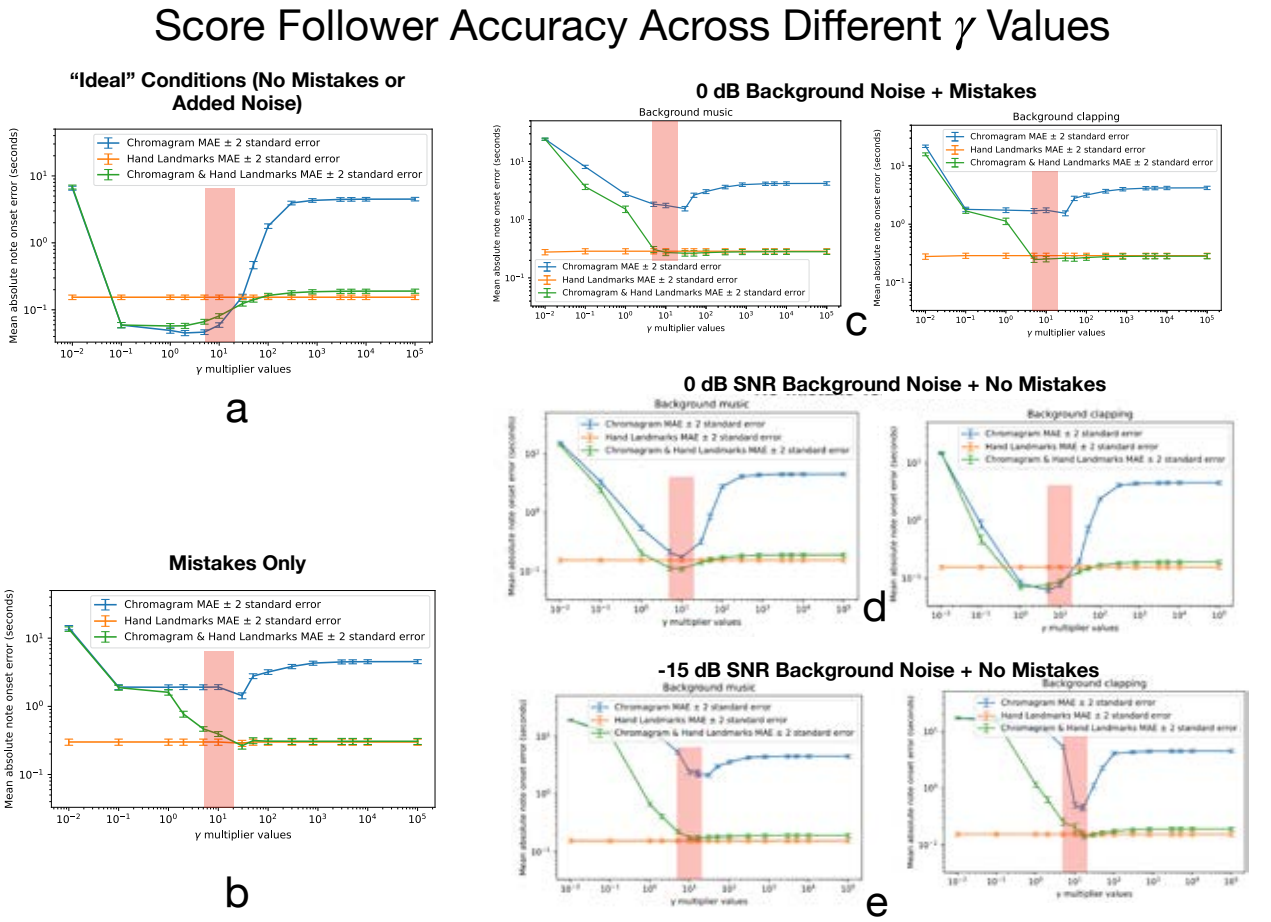


Figure 6. Highlighting behavior of varying γ in the audio-only and fused score followers, under different noise and mistake conditions. Error bars are +- 2 standard errors of the mean. In general, the combined score follower (green) exhibits more chromagram-like behavior at low penalty values and more landmark-like behavior at high penalty values. A potential "sweet spot" for γ is highlighted in red – using this range of γ values, the fused score follower always behaves closer to the better single-source score follower or outperforms both single-source score followers, despite differing amounts of background noise and the presence/absence of performer mistakes.

3.3.2 Effect of Noise on Score Following

We evaluated the score followers as different noises were added at different levels to the piano performance audio. We prepared two kinds of noise signals: (1) background music audio [14] and (2) background clapping audio. Both noises were tested at a signal-to-noise ratio (SNR) of -30 dB to 30 dB in 15 dB increments. Clapping audio was meant to simulate clapping during a live show, while music audio was intended to simulate a "backstage" or "practice room" environment where alternate music sources cannot be isolated from the input audio. Figure 4 visualizes the effect of different types and amounts of background noise with piano performance audio.

A violin plot of absolute error pieces in the *no-mistake* dataset with $\gamma = 10$ is shown in Figure 3, and the mean absolute onset error in Figure 5. Figure 6d and Figure 6e highlight behavior at 0 dB and 15 dB SNR across different γ values. Figure 5 shows how while the audio-only score follower had the best performance at ≥ 15 dB SNR, it became significantly worse than the other score followers when audio quality degraded to ≤ -15 dB SNR. As SNR decreases, performance with background music degraded more quickly than performance with background clapping. This is expected because another music signal is more likely to confuse the pitch-based audio score follower by giving a high emission probability in incorrect score states. The landmark-only score follower had an (invariant) mean absolute error of $\sim$150 milliseconds.

These results show that when the noise level varies or is unexpected, it is important to fuse the streams to take advantage of the more useful modality. In most cases, the fused score follower had a note onset error between the audio-only and landmark-only score followers. At $\gamma = 10$, it tended to perform more similarly to the source with lower error. In high-noise situations, the fused score follower performed more similarly to the landmark-only score follower, while in low-noise situations, the fused score follower more closely tracked the behavior of the audio score follower. Occasionally, the fused score follower performed better than either the audio or landmark score followers alone (for example, 0 dB SNR background music).

3.3.3 Mistake vs. No Mistake Conditions

The behavior of the different score followers on the *mistake* versus *no-mistake* datasets can be observed by comparing Figure 6b with Figure 6a. The presence of mistakes severely degrades the performance of the audio-only score follower. Without mistakes, the audio score follower was able to achieve mean absolute error of below 50 milliseconds, a level of error potentially tolerated within certain real-world score following applications. With mistakes, the average error never fell below 10 seconds – this is excessive error for any real-world score following application. Our landmark-only score follower and combined score follower showed much better resistance to performer error than the audio-only score follower. The landmark-only score follower performed comparably with and without performer mistakes – without mistakes, the average error was $\sim$150 milliseconds, while with mistakes, the average error was only raised to $\sim$290 milliseconds. The mistake-resistance shown by the landmark-only score follower is likely a result of how we approached determining ϕ from the tuning set as described in Section 3.2. Figure 6c shows how this mistake-resistance is retained even in the presence of background music or background clapping.

4. CONCLUSION

Most score followers currently rely on audio as their sole input source. Our work highlights the fragility of this strategy. In "good" audio conditions, *i.e.*, when background noise is low or the performer executes a take without extensive errors, the performance of an audio-only score follower is hard to beat. However, disruptions like background noise and mistakes can quickly derail a score follower or cause catastrophic failure if they are not specifically accounted for in the model.

We presented a multi-modal score following system that fuses deep learning-based pose features with audio. We demonstrated how this system mitigated the effect of noise and mistakes, and could be configured to harnesses the low error the audio-based score follower under "good" audio conditions and the noise/mistake robustness of a hand landmark-based score follower under "bad" audio conditions. We further showed that our hand tracking method is robust enough to be used in different video conditions – the combination of a generic hand-landmarker model and rehearsal-based training allowed successful tracking in recordings made in separate spaces, with different performers and pianos. We believe our work expands the applicability of piano score following in noisy environments, such as practice rooms or music genres for which cheering or clapping during a performance is the norm.

In the future, we hope to continue investigating how this system performs in less ideal lighting conditions that more closely simulate a live concert scenario. We also hope to investigate methods that allow for training the landmark-based features via piano fingering annotation [15], instead of a rehearsal take.

Acknowledgments

We thank Christopher Raphael for his helpful insights and letting us use his *Orchestra* program for note onset labeling.

5. REFERENCES

[1] R. Dannenberg, "An on-line algorithm for real-time accompaniment," in *Proceedings of the International Conference on Computer Music (ICMC)*, 1984, pp. 193–198.

[2] B. Vercoe, "The synthetic performer in the context of live performance," in *Proceedings of the International Conference on Computer Music (ICMC)*, 1984, pp. 199–200.

[3] A. Maezawa and K. Yamamoto, "MuEns: A Multimodal Human-Machine Music Ensemble for Live Concert Performance," in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. Denver Colorado USA: ACM, May 2017, pp. 4290–4301. [Online]. Available: https://dl.acm.org/doi/10.1145/3025453.3025505

[4] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." in *International Computer Music Conference (ICMC)*, Belfast, Ireland, Aug. 2008, pp. 33–40. [Online]. Available: https://hal.inria.fr/hal-00694803

[5] C. Raphael, "Music Plus One and Machine Learning," in *Proceedings of the Twenty-Seventh International Conference on Machine Learning*, 2010, pp. 21–28.

[6] T. Mizumoto, A. Lim, T. Otsuka, K. Nakadai, T. Takahashi, T. Ogata, and H. Okuno, "Integration of flutist gesture recognition and beat tracking for human-robot ensemble," *Proc of IEEE/RSJ-2010 Workshop on Robots and Musical Expression*, 11 2010.

[7] M. Ritter, K. Hamel, and B. Pritchard, "Integrated multimodal score-following environment," in *International Computer Music Conference*, 2013. [Online]. Available: https://api.semanticscholar.org/CorpusID:799716

[8] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee, W.-T. Chang, W. Hua, M. Georg, and M. Grundmann, "Mediapipe: A framework for building perception pipelines," 2019.

[9] Y. L. Wan, Z. G. Wu, R. H. Zhou, and Y. H. Yan, "Automatic transcription of piano music using audio-vision fusion," in *Measurement Technology and Engineering Researches in Industry*, ser. Applied Mechanics and Materials, vol. 333. Trans Tech Publications Ltd, 9 2013, pp. 742–748.

[10] X. Wang, W. Xu, J. Liu, W. Yang, and W. Cheng, "An audio-visual fusion piano transcription approach based

on strategy," in *2021 24th International Conference on Digital Audio Effects (DAFx)*, 2021, pp. 308–315.

[11] J. Lee, B. Doosti, Y. Gu, D. Cartledge, D. Crandall, and C. Raphael, "Observing pianist accuracy and form with computer vision," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2019, pp. 1505–1513.

[12] "Logic Pro X." [Online]. Available: https://support.apple.com/guide/logicpro/welcome/mac

[13] C. Raphael, "A hybrid graphical model for aligning polyphonic audio with musical scores," in *International Society for Music Information Retrieval Conference*, 2004. [Online]. Available: https://api.semanticscholar.org/CorpusID:1949303

[14] L. v. Beethoven, "Symphony No. 5 in C Minor," IMSLP, 2002, Fulda Symphonic Orchestra.

[15] W. Gao, S. Zhang, N. Zhang, X. Xiong, Z. Shi, and K. Sun, "Generating fingerings for piano music with model-based reinforcement learning," *Applied Sciences*, vol. 13, no. 20, 2023. [Online]. Available: https://www.mdpi.com/2076-3417/13/20/11321

IEMI: An Immersive, Evolving Motion-Controlled Installation

Ahmet Emin Memis (ahmetem@student.imv.uio.no) [1] and
Tejaswinee Kelkar (tejaswinee.kelkar@imv.uio.no) (0000-0002-2324-6850) [2]

[1,2] *Department of Musicology*, **University of Oslo**, Norway

ABSTRACT

Interactive Evolutionary Algorithms (IEA) are an extension of Evolutionary Algorithms, where the optimization process is centered around a persons subjective evaluation. IEAs add another layer of user interaction to transform the evolutionary process from being solely dependent on objective selection to incorporating real-time subjective evaluations. This project presents an interactive installation that allows users to engage with a sound bank by moving in space, where the mapping is updated using an evolutionary algorithm. Interactive feedback from the user is captured through button press. In this paper we present a two-stage iterative evaluation of the system. First, a pilot user study was conducted which questions the system's usability and motion capture quality, as well as making qualitative measurements. The results suggest that the system provides a new and engaging way for users to interact with their space. The system created in this paper is source open and available in a GitHub repository.

1. INTRODUCTION

Creating immersive installations is a challenge with very specific constraints. The need for making sound installations in museums and exhibitions interactive on the one hand has to be balanced with the subtle nature of the soundscapes and their sonic qualities. Overt interactive patterns exhaust the user and the soundscape, whereas repetitive sound patterns cause listener fatigue. In this paper, we explore interactive soundscapes with the help of Interactive Evolutionary Algorithms (IEA) and motion capture.

Evolutionary Algorithms are a subfield of Artificial Intelligence that uses mechanisms inspired by biological evolution. They are designed to solve problems by iteratively improving a population of potential solutions according to defined criteria [1]. Interactive Evolutionary Algorithms (IEA) are an extension of Evolutionary Algorithms, where the optimization process is centered around a person's subjective evaluation [2]. IEAs add another layer of user interaction to transform the evolutionary process from being solely dependent on objective selection to incorporating real-time subjective evaluations. IEAs make evolution processes user-centric, answering to various needs of design processes. However, this approach necessitates the user to continuously be responsive to the interface, especially in live performances. This has previously been solved by tightly sampled questionnaires, preference feedback using input tools and more. Tight sampling of user feedback creates fatigue in responding, and fails to provide seamless interaction. The metacognition of remembering to give feedback is a barrier to experiencing flow and immersion in a musical context.

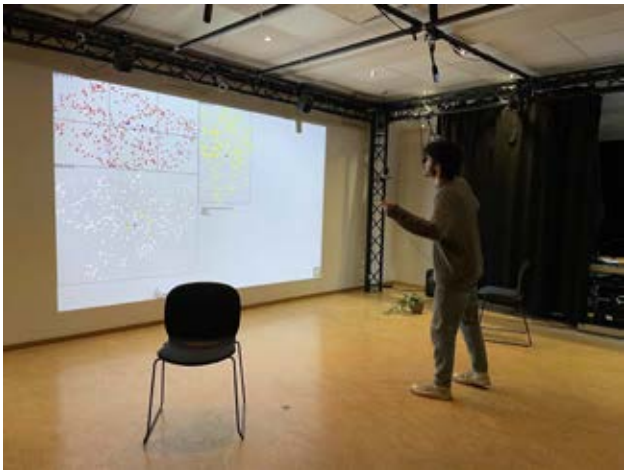


Figure 1. A user interacting with the IEMI

We present IEMI: a movement-based evolutionary algorithmic composer — a system that works with optical infra-red motion capture data as the movement feedback, where the sound is redistributed and mapped to the capture space in which the user can interact with the sound by moving their body. We present two user studies to analyze the interactive potentials of this family of algorithms in creative sound design. We make a case for movement-related feedback to shape the behavior of the system in complex and interactive ways. We see a potential for this work to contribute to the field of interaction design in sound installation.

The source code for this project is available in a GitHub repository [1] .

Recent advancements in motion capture and audio processing technologies have led to the emergence of new forms of musical instruments that enable users to engage with their bodies in novel ways [3–5]. This project aims to implement an installation that allows users to navigate sound-banks in real-time using their body movements. The controls of the installation are based on position in the

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] https://github.com/aememis/IEMI

space, quantity of motion, and parameters that are governed by an evolutionary algorithm. These parameters are mapped to various effects including delay, tremolo, reverb and frequency filtering. We use an Infra Red (IR) motion capture technology, hereafter IR mocap, to track the body movements of the user and processing of the voice using the extracted motion values. A reflective marker was placed on the body, and its positions was live streamed to be converted to motion amounts in the Python environment, then to Pure Data software where the audio processing operations were performed based on pre-calculated values. The vocal input is received through a wireless headset microphone and directed to Pure Data [2].

This project contributes to the field of human-computer interaction by exploring the potential of using motion capture and IEA technologies to create more embodied musical experiences. By enhancing the user's sense of embodiment, the vocal instrument offers new possibilities for vocal expression and creativity. The following sections of this paper provide a more detailed explanation of the methodology, results, and a discussion of the implications and potential future directions of this work.

2. RELATED WORK

With new developments in hardware, software, and human-computer interaction there is a growing interest in designing interactive systems that engage users in novel ways and enhance their cognitive and emotional experiences. Research on theoretical and experimental methods of extending artistic outcome of the body instrument has been conducted in the field of embodied music cognition. Tanaka and Donnarumma [6] explored the notion of the body as a musical instrument and argued that the body has the potential to be seen not only as a tool for producing sound but as an instrument in its own right, with its own unique affordances and sonic qualities. Jensenius and Wanderley's [7] explorations on extending the body instrument into the realm of movement-based performance art aligns closely with this work's principles. Through the integration of sensor technology and computer programming, Jensenius discusses that the body can be transformed into an instrument that generates new sounds and visual displays based on the performer's movements. Additionally, Mainsbridge [8] offers valuable insights into the use of motion-controlled performance systems to create live performances that reflect the unique style of each musician.

In this work, we take inspiration for designing interaction from several works in which position, posture and movement are mapped. One particularly influential model is OtoKin [9], which introduces a collaborative performance space abstracted in 3 dimensions and divided into specific zones for sound controls. Additionally, Fornari's application of evolutionary algorithms for sonic landscape design aligns closely with our strategy to build the auditory environment in response to user interaction [10]. They used these algorithms to generate birdsongs to build an artificial sonic landscape. Furthermore, they discuss the feasibility of integrating interactivity for fitness in their proposed system. To explore the integration of adaptivity within interactive music systems, Erdem et al.'s CAVI [11] provided valuable insights. Their work discussed the development of an interactive system that is as much an adaptive composition system as an interactive musical instrument. In this context, they presented the CAVI, a co-adaptive instrument that generates audiovisual responses dynamically evolving in response to user interaction.

2.1 Research Objectives

Drawing inspiration from this related work, we aim to create a motion-based interactive system for exploring evolutionary algorithms. Our main research objective is to learn

> does using motion capture for user feedback create an immersive sound experience to interact with an evolutionary algorithm.

To this end, we created a prototype for the algorithm, and studied interaction behavior with this algorithm. We are interested in how this user interaction is able to create a concept of control of sound parameters. Since the feedback is both motion based, and unary, we are interested in how the gradually changing slow interaction is experienced by people interacting with such a system.

3. DESIGN AND IMPLEMENTATION

The design focus for IEMI is to allow the user to control music creation through their body movements and spatial interaction, while experiencing the auditory results to provide real-time feedback that contributes to the evolution. Our system comprised an interactive space equipped with motion capture and spatial audio technology, alongside a computational framework that employs evolutionary algorithms.

Both prototypes utilized parameters steered by the evolutionary algorithms, projected into three-dimensional space for user interaction. This setup allowed users to navigate and explore with the parameter space through physical movement. The auditory outcome dynamically reflects the user's position within the environment, based on a proximity-based selection of parameters where the nearest data point in the space takes precedence.

Our interaction design focus was creating an interactive space in which the user can freely move and sound-walk to explore and adapt the space based on their musical preference. To achieve this, we integrated motion capture to track the user's motion and a feedback mechanism to gather real-time feedback from the user. The user navigated the interactive space using the marker, with the system processing a continuous stream of the marker's position as input. For giving subjective feedback, we used a computer mouse where a click represented a *like*.

In both prototypes, our sound design aimed to create an engaging auditory experience that showcased the adaptive behaviour of the sound installation. To achieve this, the

[2] https://puredata.info/

initial prototype was designed to manipulate effect parameters on vocals and continuous audio samples, while the second prototype introduced the dynamic selection of audio samples from a comprehensive sound bank. Figure 2 illustrates the main components of the system.

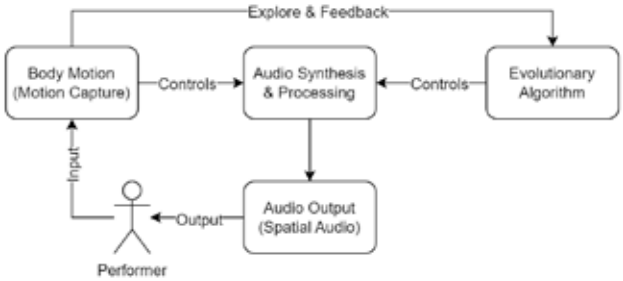


Figure 2. Main components of the IEMI system.

3.1 Equipment

To capture the user's body motion, we used Optitrack's infrared motion capture system with 8 cameras [3]. The voice input was captured using a wireless headset microphone. The output was played through a rectangular array of twelve speakers which were suspended from the ceiling. Finally, the visual content was projected to a wall of the room, which can be seen in Figure 4.

3.2 Motion Capture

To track the user's location, we used a reflective marker controlled by the user. The real-time positional data of this marker was continuously streamed from a stationary computer to Python environment using NatNet [4], an SDK capable of cross-platform motion capture data streaming. Within this environment, we obtained the marker's exact position in three-dimensional coordinates.

The motion intensity was quantified by processing the position data stream, specifically by calculating the Root Mean Square (RMS) over a sliding window of the last 100 samples. This provided a dynamic measure of user's activity within the space. For the design of P1, which required segmentation of the room floor into distinct zones, the interaction space was divided into six virtual zones, and the specific zone occupied by the user was continuously identified in real-time.

3.3 Data Representation and Preprocessing

We designed the individuals in the population to represent certain characteristics of the auditory feedback. These included effect control parameters for P1, while being features representing the audio samples for P2. These parameters are stored as an array of floating point numbers, scaled between 0 and 1.

To facilitate room-scale interaction with the population through motion, dimensionality of the population was reduced and projected into a three-dimensional space. We employed Uniform Manifold Approximation and Projection (UMAP) algorithm for this purpose, which is known for its ability to preserve more of the global structure compared to other well-known dimensionality reduction techniques [12]. UMAP also served as an effective clustering tool by grouping similar data points and assigning distinct characteristic sounds to different areas in the room. This eliminated the need for users to search randomly. During this process, direct link to the original population was preserved. This approach ensured that interactions with the UMAP representation could directly influence the original population. The UMAP representation was scaled to match the dimensions of the physical room that the installation took place.

As the user navigated the space with the marker, the nearest individual was determined through ongoing calculation of Euclidean proximity in three-dimensional space. The individual nearest to the user's current position was deemed the active selection. The active individual took precedence and controlled the ongoing sonic outcome. Moreover, the feedback received from the user was directly associated with the corresponding individual in the population. These projection approaches enabled effective interaction with high-dimensional populations through a scaled-down projection. Figure 3 shows the continuous transition between the UMAP projection and the population.

Incorporating these calculations, a real-time visualization of the motion capture interaction was displayed to the users, aiming for increased engagement and understanding of the system's response to their movements. Figure 4 shows a screenshot of this visualization during the interaction. It illustrates the spatial layout of the UMAP projection within the interaction zone across three planes. Additionally, it shows the six virtual zones, the user's position, 5 nearest neighbours, closest neighbour, and additional contextual information like current position in coordinates and elapsed time.

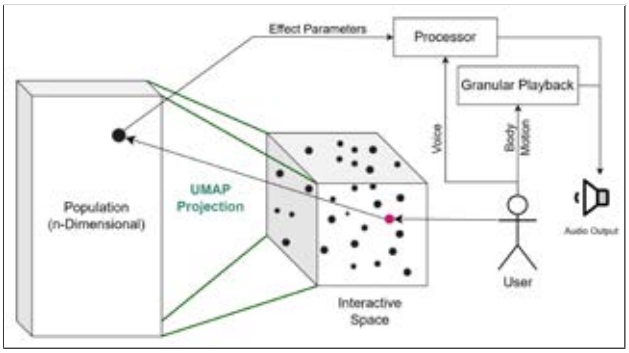


Figure 3. The system selects the nearest point within the 3D space based on user's position, identifies the corresponding individual in the population, and activates it to control the audio effects. Simultaneously, user's body motion controls the granular playback.

[3] https://optitrack.com/
[4] https://optitrack.com/software/natnet-sdk/

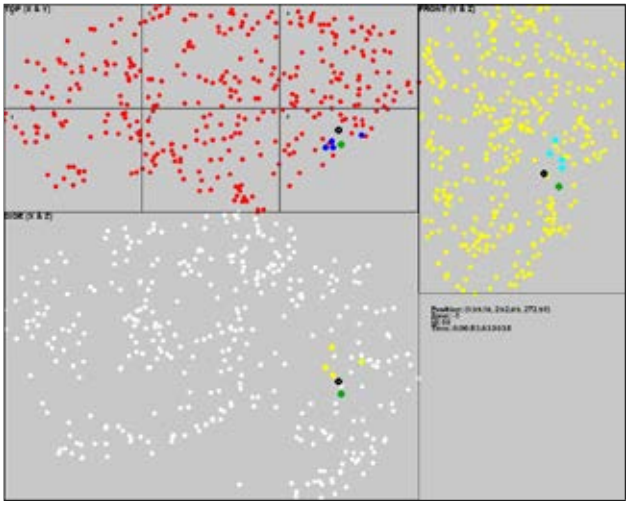


Figure 4. A screenshot of the visualization shown to the user, highlighting the UMAP projection's distribution within the interaction space. A black dot represents the marker's location and a green dot marks the nearest individual, and six virtual zones are shown within the TOP view.

3.4 First Prototype - P1

The first prototype of IEMI was designed to provide an engaging auditory experience using user's vocals, and audio samples. In the core of the prototype, the user sings while controlling sound synthesis and various audio effects are applied to their voice and the audio samples. P1 allowed the user to create the auditory outcome through their voice and body movements—either directly influencing the sound or indirectly, via the evolutionary algorithm, to guide the system's adaptive responses.

The main sonic elements of prototype P1 were the vocal input and a set of audio samples. The vocal input is captured using a headset microphone. Meanwhile, the audio samples, including both ambient sounds and vocal elements, were played back in grains.

The evolutionary algorithm parameters controlled the audio effects with a complex mapping. Initially, the installation began with a randomly generated population. This population then underwent evolution, driven by real-time feedback from the user. Firstly, similar to [9], we introduced a "sound-walk" concept where the floor of the room was divided into six zones, allowing the user to use these zones as a control method. As the user stepped into a zone, playback of pre-assigned audio samples was triggered. These samples comprised of open source recordings of a steel hang instrument, speech, whispering, and layered acapella singing. The design intent behind this selection was to craft an immersive soundscape containing several spectral, and envelope characteristics. As the primary focus of this prototype was on vocals, we sourced mainly vocal material to maintain a consistent theme. The zones also controlled other parameters in the sound generation, such as carrier frequency of drone and grain lengths.

The spatial audio distribution was linked to the user's orientation with respect to the x-axis, with variable rules applied to different samples enrich the sonic experience. This approach applied direct or modified correlation to the user's orientation to sound distribution, introducing a nuanced responses to the sound distribution. Moreover, moving away from the center resulted in a more focused sound output from a narrower range of speakers, whereas moving towards the center produced a broader sound.

User Interaction	Control parameter
Quantity of Motion	Volume
Virtual zone	Triggering audio samples
Position	Spatial distribution
Height	Grain speed
Population Parameters	Delay, Tremolo, Reverb and Frequency Filtering

Table 1. Mapping of user interaction to control parameters.

This prototype was designed to respond dynamically to a user's physical activity by establishing a direct relationship between the quantity of motion and the auditory feedback. Specifically, standing still would result in increased volume, while more movement would lead to reduced volume. This design choice aimed to encourage the user to pause and immerse themselves with the auditory stream. Moreover, it refined the feedback mechanism, ensuring that interactions were accurately attributed to the most recently engaged individual within the evolutionary parameter space. Finally, to further enrich the overall experience, we integrated continuous drone sounds and other ambient elements. Table 1 shows the mapping between motion and audio processing.

Necessary calculations were made either in Python, or Pure Data, a visual programming language tailored for audio processing. The parameters calculated within Python environment are transmitted to Pure Data for the sound synthesis and further processing.

3.5 Second Prototype - P2

The second iteration focused on exploring a sound collection to create an immersive and adaptive dynamic auditory landscape. To achieve this, we used a large dataset of audio samples, and an evolutionary algorithm that processes audio features extracted from these samples. The virtual space was designed to evolve by gradually selecting audio samples that aligned more closely with user preferences.

We used the *Freesound Dataset 50 k* [13], a large dataset of short audio clips. From this dataset, we randomly selected 4000 audio clips, each with durations ranging from 0.25 to 4 seconds.

To prepare the data for integration into the installation, we focused on capturing their characteristics through calculating a set of features. This was achieved by analyzing the clips within sliding windows, and computing the mean values of several key audio features. These features were chosen strategically to represent the clips' dynamic range, timbral quality, and texture. These features included Root Mean Square for intensity, Spectral Bandwidth and Centroid for timbral characteristics, Spectral Flux for sound

texture changes, and Mel-Frequency Cepstral Coefficients for unique sound signatures, along with Spectral Flatness to measure tonal versus noisy sounds. The length of each clip was also recorded to understand the temporal aspect.

The evolutionary process was driven by the extracted audio features to ensure that the population of audio samples could evolve in response to user feedback. Each individual within the space represented a unique set of audio features. As the user interacted with the environment, the system continuously identified and selected the audio sample whose audio features most closely matched those of the individual currently being engaged by the user. To identify the most similar audio sample, we calculated audio features' Euclidean distance. This approach allowed the system to adjust to user preferences by introducing new audio samples from the dataset that best matched their interests. Figure 5 illustrates this process.

3.6 Algorithm Design

Evolutionary behavior in IEAs is mainly guided by the principles of genetic algorithms, which mimic biological evolution through the creation and optimization of solutions via selection, mutation, and crossover. The fitness of the audio samples is determined by user feedback, which makes the process interactive and allows for adaptation in response to user preferences.

For both prototypes, the algorithm initiated with a population size of 350. In the case of P1, this consisted of randomly generated parameters, whereas for P2, it involved a random selection of audio features from the dataset.

The performer is granted the freedom to navigate the 3-dimensional UMAP representation of the population and engage with and react to the auditory results. Navigating such a complex space effectively poses a significant challenge and makes it nearly impossible to construct a mental model of their interactions [14]. Therefore, similar to the strategy suggested by [14], the system was designed to be explored in shrinking steps. In this iterative approach, evaluation process progresses with only a portion of the population being assessed. Subsequently, the system evolves by generating individuals that resemble the preferred solutions. This search method, although it may deviate from the Darwinian evolutionary approaches, facilitates faster convergence, while maintaining the control over the direction of the process.

For each individual in the population, the performer's feedback is unary, recorded as "liked", or left "not liked or encountered" (e.g., 1 or 0). Once the portion of the rated individuals exceeds a certain threshold, this triggers the selection process. While deciding on this threshold, one key factor was ensuring that the number of individuals resulting from crossover and mutation processes closely matches the size of the initial population. The selection process selects the rated individuals for producing offspring, while the unrated ones are eliminated in the next generation.

Following the selection, the individuals progress with the variation operators, specifically crossover and mutation. In the crossover phase, for each pair of individuals, a randomly chosen segment from one partner is exchanged with the corresponding segment from the other partner. The offspring produced by this exchange is passed to mutation phase, where the randomly chosen parameters within individuals are altered by a random factor. The probability of mutation was set to 0.5. To enhance diversity, the mutated individuals are added as new entries for subsequent generations. Figure 6 shows the flow of the interactive genetic algorithm.

4. EVALUATION

We developed IEMI in two iterations. The first prototype, P1 was evaluated with the help of a purely qualitative pilot study. The interview questions pertained to immersion, interaction and perception of control, and free communication about the general experience of the system.

The evaluation sessions involving participants were conducted in accordance with the ethical guidelines by the Norwegian Agency for Shared Services in Education and Research. Participants consented to participation anonymously and their interactions with the system were not recorded in any form. Additionally, their responses to interviews and questionnaires were collected without any personal data.

4.1 Pilot Study

The first iteration of the algorithm was tested in a pilot study. The aim of this study was to obtain a more free feedback from the participants about the interaction, and learn the patterns of interaction that are interesting for people. There were 5 participants in the study, 3 women and 2 men with ages ranging from 28 to 61. The study participants were all musicologists from the University of Oslo, with 10+ years of training in music. In 10-minute long interviews, we asked the participants about various kinds of feedback regarding both the motion as well as vocal interaction. Not everyone was interested in testing vocalization, and many commented on the nature of the task of vocalizing to not fit the purposes of a sound installation. Since vocal datasets were included in this first iteration, one participant also remarked that they were unable to distinguish at times if the vocal sounds they heard were previously produced by them.

Each of the 5 participants in this study remarked on thinking that the sound-bank was what was evolving, and not the effects. We realized that the soundbank like aspects of the design take precedence in the soundscape due to their salience. Effects-based sound design makes it harder to distinguish between changes and evolution. This creates an expectation in the participants to *want to hear* changes in soundbanks. Thus, we concluded that a system directly playing sound samples, rather than synthesizer-generated sound, would offer users clearer interaction and enhance the evolutionary changes over time. Following the results of this study, it was decided that P2 - the second prototype, would be based on a soundbank-based design.

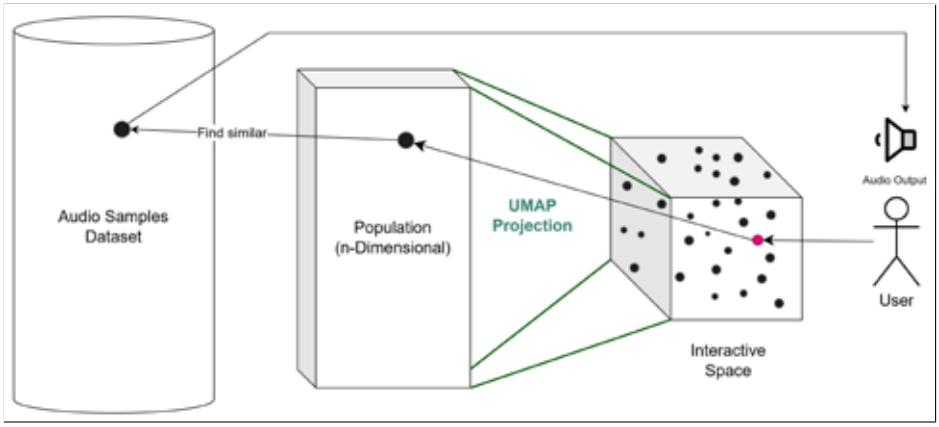


Figure 5. The system selects the nearest point within the 3D space based on user's position, determines the corresponding individual in the population, and plays the most similar audio sample from the dataset. (P2)

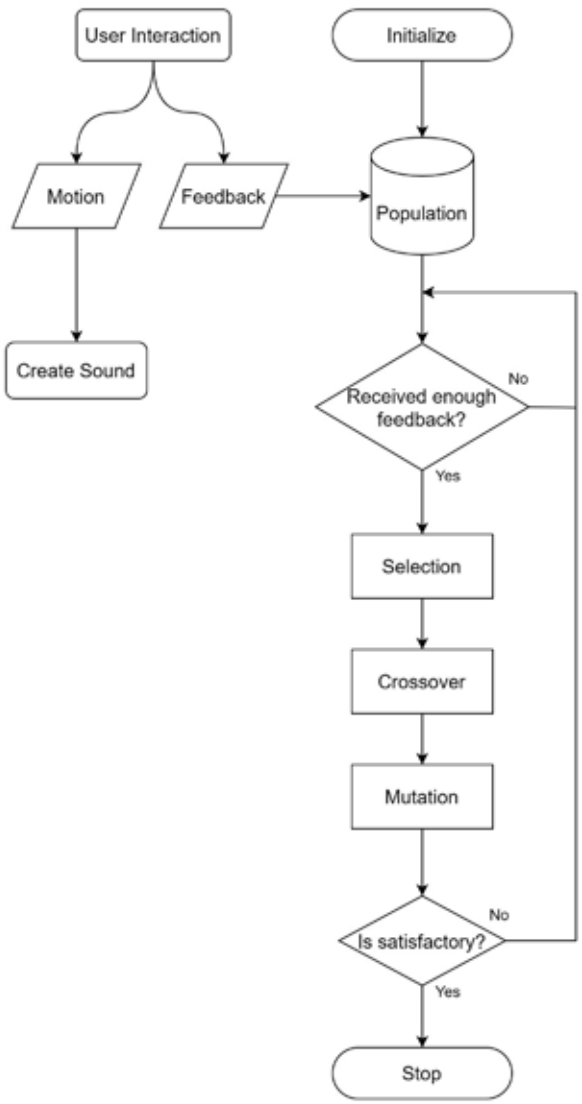


Figure 6. Flowchart of the IEMI Interactive Genetic Algorithm

4.2 Follow-Up Study

The follow up study consisted of 11 participants, with an average age of 30. The follow-up study was divided in two parts. In the first part, the participants interacted with the first prototype P1, while in the second they interacted with the second prototype P2. We were interested in capturing the differences in their experiences based on the first iteration of the experiment.

In this group, the range of years of musical training was more varied, with 45% of the participants having had less than 3 years of musical training and 45% having had over 10 years of musical training. Coding their feedback was based on the following categories of responses, each aligned with specific questions asked after the sessions:

- Degree of immersion — Did you feel immersed in playing with the installation?
- Degree of liking the interaction — Did you like interacting with the sounds?
- Degree of experience of control — Did you feel like you had control over the changes in sound?
- Degree of understanding how it works — Did you understand how the sound and movement worked together?
- Degree of experiencing change — Did you feel like the system changed over time?
- Degree of experiencing control over change — Did you feel like you had control over the change in the system?
- Degree of liking changes to the soundscape — Did you feel like the system changed in a way that you liked?

One participant remarked:

> *The spatial interaction was interesting, the second experiment was more immersive than the first one. Moving my arms allowed me to control the sound to a great degree and that was an engaging creative experience.*

Controlling sounds in space can become an inviting and exploratory task, where not restricting oneself to the interfaces of buttons, screens, or tools frees people to move as they like. Another participant here remarked:

> *I felt like the system could be played in a variety of ways which greatly influenced the overall experience in interesting ways.*

Participants remarked that the system became significantly more interesting once they realized that standing still could be used as a way of controlling the system.

4.3 Results

We focus on user feedback relating to qualitative and experiential aspects from the users. Out of the 16 participants involved in both P1 and P2 testing, more than half reported that they had fun using the system, and enjoyed navigating the soundscape through movement.

In the Follow-Up user study, we found that there was a greater degree or invitation to interact with the P2 system than P1. There is a large impact of the sound banks on the results of the surveys in these tests. As the surveys stand at the moment, it is impossible to separate the effects of sound design from the preference-related aspects of particular sounds that people like and dislike within the sound installation. The differences in the choices of soundbanks made a huge difference to the interaction of the system. As such, operationalizing this system with any soundbank is a possibility. Navigating point-clouds in 3D space with a target marker with visual as well as auditory feedback has potential in creating engagement and immersion.

5. DISCUSSION

In this paper, we described a prototype instrument for interacting with evolutionary algorithms controlling sound. We demonstrated two prototypes with a pilot user study and a final user study, indicating some factors that influence perception of control through unary feedback, and how people relate to this type of algorithm design. However, the two stand-out factors from these studies were in the "like-only" interaction paradigm, or unary feedback, as well as the perception of control.

5.1 Unary feedback for Engagement

How to obtain user feedback from participants in sound installations is a question that is even harder to answer at scale. This type of a question may also have different answers in different performance contexts. For example, in the context of a museum, engagement means moving as little as possible, whereas in the context of a club, maximal motion indicates maximal engagement. In the development of these tools, we considered these to be unary feedback, where negative feedback by the user is hard to capture. The system design was therefore based on a one-way feedback mechanism. We note that this type of feedback is quite common in popular algorithmic interactions on social media, where there has been no way to indicate displeasure and disengagement with content, and non-engagement can be interpreted both as indifference as well as disliking something. Some developers refer to this type of feedback as "Unary Implicit", where the data is similar to binary data, but in the real world the data is better represented as integers, since the 0 value of such a dataset is overloaded with both disliking, or never encountering something.

5.2 Perception of Control

When moving in space in order to control sound is established as a baseline condition within a task such as this, users can sometimes perceive controlling the sound when there is no control or change. Aspects of this are explored by various authors writing about agency and control in NIMEs [15–17], but we think this instrument prototype can be used to explore this further. Most participants in a study like this remark wanting to understand the system, and their interest in solving the puzzle of sonic interaction. The relationship between moving less in order to create a louder sound was perhaps the most elusive one for creating an illusion of control.

The results of the experiments suggested that the installation was capable of providing an intuitive musical experience for users with varying levels of musical background and age. They also provided valuable feedback on the system, highlighting areas for improvement such as a looper functionality or having a more musical case as the core of the sound design such as a drum sequencer.

In the qualitative interviews, study participants commented on their perception of control of the system. Since IEA-based systems take some time to evolve, the effects of the interaction are not immediately available for participants. In that way, change is both gradual as well as subtle. Participants liked the subtlety of this change, and got used to the time scale at which feedback occurred. However, this is both an advantage as well as a limitation while designing sight-specific works that change gradually.

5.3 Limitations and Future Work

Both the evaluation of P1 as well as P2 is with a small sample size focusing on qualitative and experiential aspects of the system. Overall, the results of the user study provide valuable insights into the potential of motion capture technology for embodied musical performance. The feedback from participants highlights areas for improvement and future development of the system. This study is a proof of concept study for incorporating interaction in evolutionary algorithms based interactive music systems.

As future work, we would like to test this system with a statistically significant user sample. The nature of the system is such that one could replace soundbanks to navigate through datasets of their own choice in an active listening scenario.

In general, an IEAs are hard to benchmark due to the freely changing nature of the system. To benchmark this, we plan to conduct systematic study using the same path and different algorithmic conditions to understand how

systematic change can be measured in IEMI, and other IEAs.

Acknowledgments

We thank all the participants who generously shared their valuable feedback and gave us their time to interacting with this soundscape.

6. REFERENCES

[1] A. E. Eiben and J. E. Smith, *Evolutionary Computing: The Origins*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 13–24. [Online]. Available: https://doi.org/10.1007/978-3-662-44874-8_2

[2] H. Takagi, "Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation," *Proceedings of the IEEE*, vol. 89, no. 9, pp. 1275–1296, 2001.

[3] A. O. Effenberg, U. Fehse, G. Schmitz, B. Krueger, and H. Mechling, "Movement Sonification: Effects on Motor Learning beyond Rhythmic Adjustments," *Frontiers in Neuroscience*, vol. 10, p. 219, 2016.

[4] G. Dubus and R. Bresin, "A Systematic Review of Mapping Strategies for the Sonification of Physical Quantities," *PLOS ONE*, vol. 8, no. 12, p. e82491, Dec. 2013, publisher: Public Library of Science. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0082491

[5] A. Aska and M. Ritter, "Approaches to Real Time Ambisonic Spatialization and Sound Diffusion using Motion Capture," Jan. 2016.

[6] A. Tanaka and M. Donnarumma, "The Body as Musical Instrument," in *The Oxford Handbook of Music and the Body*, Y. Kim and S. L. Gilman, Eds. Oxford University Press, Oct. 2019, p. 0. [Online]. Available: https://doi.org/10.1093/oxfordhb/9780190636234.013.2

[7] A. Jensenius, M. Wanderley, R. Godøy, and M. Leman, "Musical Gestures: concepts and methods in research," *Musical Gestures: Sound, Movement, and Meaning*, Jan. 2009.

[8] M. Mainsbridge, *Body as Instrument: Performing with Gestural Systems in Live Electronic Music*. Bloomsbury Publishing USA, Feb. 2022, google-Books-ID: wGxWEAAAQBAJ.

[9] P. Dahlstedt and A. Skånberg, "Otokin: Mapping for sound space exploration through dance improvisation," vol. Paper i proceeding 2019, 08 2019.

[10] J. Fornari, "An evolutionary algorithm to create artificial soundscapes of birdsongs," *International Journal of Arts and Technology*, vol. 9, p. 39, 03 2016.

[11] C. Erdem, B. Wallace, and A. R. Jensenius, "CAVI: A Coadaptive Audiovisual Instrument–Composition," in *NIME 2022*. The University of Auckland, New Zealand: PubPub, Jun. 2022. [Online]. Available: https://nime.pubpub.org/pub/cavi

[12] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.

[13] E. Fonseca, X. Favory, J. Pons, F. Font, and X. Serra, "Fsd50k: An open dataset of human-labeled sound events," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 829–852, 2022.

[14] P. Dahlstedt, "Evolution in Creative Sound Design," in *Evolutionary Computer Music*, E. R. Miranda and J. A. Biles, Eds. London: Springer London, 2007, pp. 79–99. [Online]. Available: https://doi.org/10.1007/978-1-84628-600-1_4

[15] G.-M. Schmid, "Evaluation of musical instruments from the musician's perspective," Ph.D. dissertation, University of Basel Basel, Switzerland, 2015.

[16] W. Brent, "Perceived control and mimesis in digital musical instrument performance," 2012.

[17] C. Erdem, Q. Lan, and A. Jensenius, "Exploring relationships between effort, motion, and sound in new musical instruments," *Human Technology*, vol. 16, pp. 310–347, Nov. 2020.

A Cover Image Composition Method for Music Content Using Multimodal Image Retrieval and Cropping Techniques

Takayuki Nakatsuka, **Masahiro Hamasaki**, and **Masataka Goto**

National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan
{takatuki.nakatsuka, masahiro.hamasaki, m.goto}@aist.go.jp

ABSTRACT

Aesthetic image compositions capture people's attention. For music content, music cover images and thumbnail images play an important role in enhancing visibility and viewer engagement of the content. While image cropping techniques can generate aesthetic image compositions from a single image, generating image compositions from multiple images for music content remains unexplored. The challenge is to ensure that these image compositions maintain a specified aspect ratio, possess high aesthetic quality, and are contextually relevant to the music content. To address this challenge, we propose a cover image composition method that generates aesthetic image compositions from an image collection and retrieves image compositions suitable for music content. The key technical aspect involves constructing a multimodal embedding space using multimodal image retrieval and cropping techniques. Within this space, feature vectors of music audio, aesthetic image compositions, and text with similar contexts can be placed closely by optimizing them using our proposed loss function. Given music content (music audio or text) as a query, our method can retrieve image compositions suitable for the query on the basis of the similarities of their feature vectors in the space. Through qualitative analysis and quantitative evaluation, we demonstrate the effectiveness of our proposed method.

1. INTRODUCTION

Visual content designed for music, such as cover and thumbnail images, serves as the initial point of engagement between music content and viewers, substantially contributing to advertising music content and enriching the viewing experience [1, 2]. Consequently, musicians devote their efforts to creating aesthetic cover (or thumbnail) images tailored to their music content. To create such images, some leverage stock images from sources like Adobe Stock [3] and Unsplash [4]. The problem is that these stock images do not always have the optimal aspect ratio required for such images, typically specified as 1:1 for cover images and 16:9 for thumbnail images by online music services. To address this problem, image cropping techniques are used to generate aesthetic image compositions in the specified aspect ratio.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

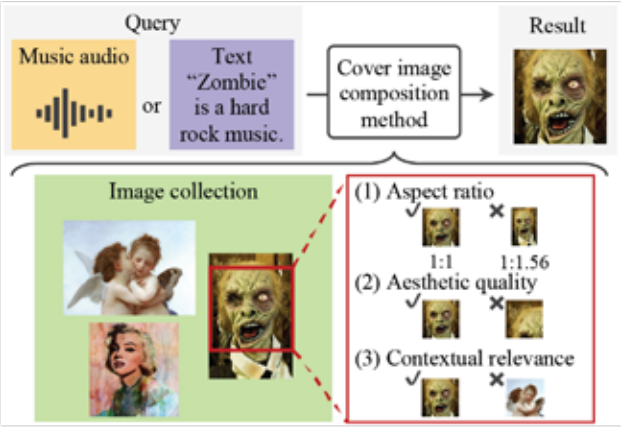


Figure 1. Conceptual design of the task in this study. Our objective is to retrieve image compositions suitable for a music audio or text query. The image compositions are generated from an image collection and fulfill three requirements: they have a specified aspect ratio, high aesthetic quality, and contextual relevance to the query.

Image cropping can identify aesthetic image compositions within an image, considering factors such as saliency, composition, and aesthetics [5]. Several studies have proposed image cropping techniques aimed at selecting specific regions or objects within an image to align with the user's intention [6–9]. However, these techniques primarily focus on cropping from a single image and do not assist users in finding images from an image collection, such as stock images. Given such a collection, generating aesthetic image compositions suitable for music content remains challenging.

In this paper, we propose a cover image composition method to address this challenge, as illustrated in Figure 1. Our method leverages music content (music audio or text) as a query to retrieve suitable image compositions. Such a query enables users to easily find their desired image compositions. Furthermore, cover and thumbnail images must be adjusted to a specified aspect ratio, possess a high aesthetic quality, and match the context of the music content. Therefore, our method takes these requirements into account when generating image compositions.

To generate such image compositions, our method utilizes multimodal image retrieval and cropping techniques. For the retrieval, we first construct a multimodal embed-

ding space in which each music audio, image, and text is represented as a feature vector by training respective encoders that embed them into the space.

To train the encoders, our method uses contrastive learning [10–12], a deep learning technique that embeds data into the multimodal embedding space so that feature vectors of similar context data are close to each other, while those of dissimilar ones are far away. By optimizing the space using contrastive learning, we can closely align the feature vectors of music audio, images, and text with similar contexts, thereby facilitating the search for image compositions on the basis of a music audio (or text) query.

Additionally, we use image outpainting and cropping techniques to ensure that the encoders identify images of high aesthetic quality. Image outpainting can seamlessly expand the boundary of an input image. We utilize the mask-aware transformer (MAT) [13] to generate the outpainted images of cover (or thumbnail) images. We then train the encoders to prioritize cover (or thumbnail) images over images randomly cropped from the outpainted images in the multimodal embedding space.

The trained encoders can be utilized to embed a music audio (or text) query and image compositions into the multimodal embedding space. We use a grid-anchor formulation [14] to generate image composition candidates from each image in the image collection. Subsequently, we can generate a ranked list of image composition candidates on the basis of the similarity between the feature vectors of the query and each candidate.

We qualitatively show the effectiveness of our method using the TAD66K dataset [15] as the image collection. In addition, we demonstrate our method's effectiveness through quantitative evaluation on the public YT8M-MusicVideo dataset [16] and the private Album Songs 5 Million (AS5M) dataset.

2. RELATED WORK

Image cropping automatically selects visually-appealing regions or objects from an image for various applications [17]. Advancements in deep learning techniques have enabled image cropping techniques to become more practical [18]. However, these techniques mainly focus on improving the aesthetic quality of the cropped images, resulting in the images that do not capture the user's intention.

Several studies have been pursuing the potential for further advancement in image cropping, particularly in capturing the user's intention through multimodal input [6–9]. Santella et al. introduced a framework that implicitly utilizes gaze-based interactions to identify accurate regions of interest [6]. Bhattacharya et al. developed a framework that recomposes an image on the basis of the user-selected foreground object [7]. Horanyi et al. proposed caption and aesthetic-guided image cropping, which leverages pretrained models for image captioning and aesthetic tasks [8]. Zhong et al. developed a framework that integrates OWL-ViT [19] and DETR [19] to achieve query-conditioned image cropping [9]. The drawback of these image cropping techniques is that they only search for image compositions in a single image. Our method can generate image compositions from multiple images, facilitating the search for image compositions the users want.

3. PROPOSED METHOD

This section describes our proposed method. Figure 2 shows an overview of our proposed method.

3.1 Data Representation

Our method deals with three modalities of data: music audio, images, and text. Here, we describe each data representation in our method.

3.1.1 Music Audio Representation

The music audio is converted to a mel spectrogram through a feature extractor of contrastive language-audio pretraining (CLAP) [20] available in Transformers [21], and our audio encoder is trained using the spectrogram as input. To train our audio encoder, we apply a masking technique [22] and a random crop technique [23] to the spectrogram for data augmentation. The masking technique generates random masks on the spectrogram in both frequency and time domains, and the random crop technique selects a random section of the music audio.

3.1.2 Image Representation

We use an RGB image resized to $224\,\text{px} \times 224\,\text{px}$ as input for our image encoder. Since most datasets for image cropping are single-modal, comprising only images, multimodal datasets that include music audio are not readily available. Hence, we create such a dataset using image outpainting and cropping techniques. Specifically, we utilize MAT [13], an outpainting technique, to expand the boundary of the original cover (or thumbnail) image. We resize the image such that its longer side becomes 224 pixels while maintaining the original aspect ratio, and then we center the resized image and use MAT to outpaint around it, resulting in an image size of $512\,\text{px} \times 512\,\text{px}$. During training, we operate under the assumption that the region of the original image represents a more suitable composition than randomly cropped regions from the outpainted image. This approach is based on the actual design workflow in which cover and thumbnail images are often created by cropping from photos or illustrations. It has the unprecedented advantage that original images actually cropped by professionals can be used as correct answers in this manner. To train our image encoder, we use both the original and outpainted images. The outpainted images undergo a series of data augmentation: a random resized crop (with scale range [0.57, 1.43] and ratio range [0.4, 2.5]), a random horizontal flip (with a probability of 0.5), and random erasing (with a probability of 0.2) [24].

3.1.3 Text Representation

We tokenized text generated by using a keyword-to-caption augmentation technique [20] with a maximum length of 77, which is the same setup as contrastive language-image pretraining (CLIP) [25]. To train our

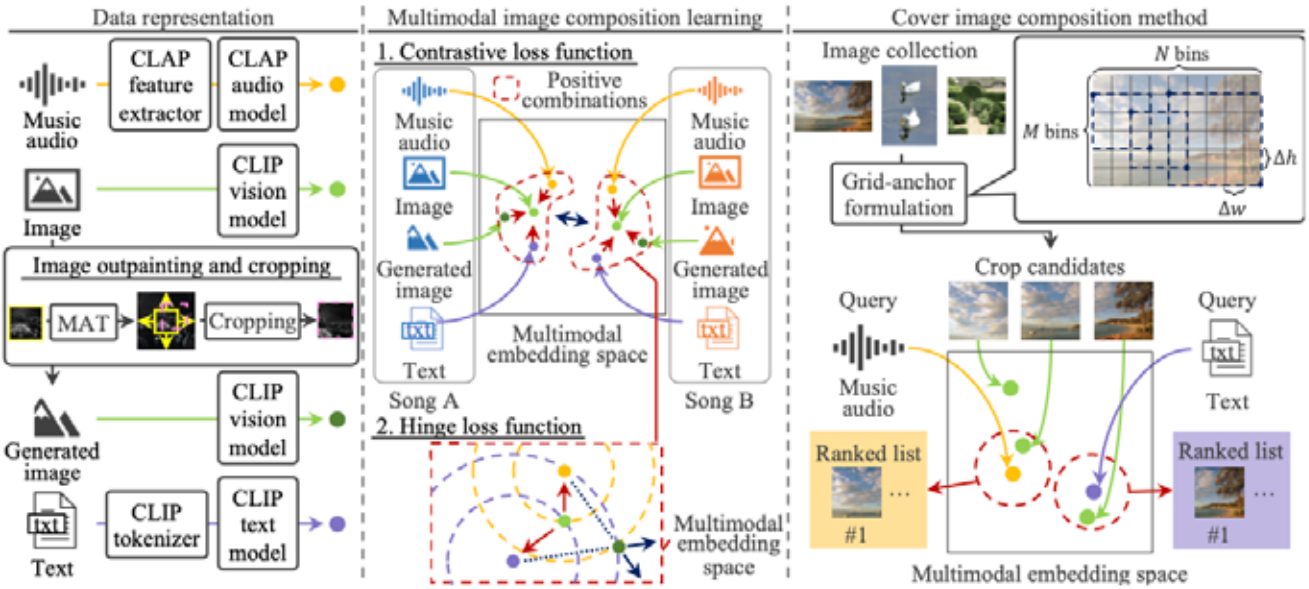


Figure 2. Overview of the proposed method that leverages multimodal image retrieval and cropping techniques. Our method utilizes music audio, images, and text. In multimodal image composition learning, we train encoders so that feature vectors of music audio, an image, and text from the same song are close in a multimodal embedding space, whereas those from different songs are far apart. Additionally, we use outpainting and cropping techniques to generate images. We also train the encoders so that the feature vector of an aesthetic (i.e., original) image is closer to those of the corresponding music audio and text than that of the generated image in the space. Once the training is completed, our cover image composition method can generate a ranked list of image composition candidates for a given query from an image collection.

text encoder, keywords corresponding to metadata are randomly dropped [26] at a ratio of 0.05 for each metadata.

3.2 Multimodal Image Composition Learning

This section describes our designs of each encoder and a learning framework to train the encoders.

3.2.1 Encoder Architecture

We use an audio model of CLAP [20] as the audio encoder, and use image and text models of CLIP [25] as the image and text encoders, respectively. We utilized pretrained models available in Transformers [21] (i.e., "laion/clap-htsat-fused" for CLAP (audio model) and "openai/vit_base_patch16_224" for CLIP (image and text models)). During training, we updated the projection layers of the encoders. Each encoder outputs a 512-dimensional feature vector.

3.2.2 Learning Framework

To jointly train multiple modalities of data, contrastive learning is an effective approach [16, 20, 25]. Inspired by contrastive loss functions that calculate the loss on the basis of N-pairs of instances (i.e., one positive pair and $N-1$ negative (or irrelevant) pairs) like N-pairs loss [10], InfoNCE loss [11], and MoCo [12], we design a contrastive loss function $\mathcal{L}_C$ that considers all pairwise combinations of music audio, images, and text as follows:

$$\mathcal{L}_C = -\frac{1}{m}\frac{1}{|\mathcal{S}|}\sum_{\boldsymbol{S}\in\mathcal{S}}\sum_{i=1}^{m}\log\frac{e^{S_{i+}/\tau}}{\sum_{j=1}^{m}e^{S_{ij}/\tau}}, \quad (1)$$

where m is a mini-batch size, τ is a temperature scaling parameter that controls the scale of the loss function, $+$ indicates a positive instance of an anchor, $\boldsymbol{S}$ is a similarity matrix whose element S_{ij} is defined as the cosine similarity $\mathrm{sim}(\cdot,\cdot)$ between an anchor and an instance in a mini-batch, and $\mathcal{S}$ is a set of similarity matrices that are constructed by using all combinations of music audio feature vectors $\{\boldsymbol{z}_n^A\}_{n=1}^m$, image feature vectors $\{\boldsymbol{z}_n^I\}_{n=1}^m$, and text feature vectors $\{\boldsymbol{z}_n^T\}_{n=1}^m$. While training, we also utilize feature vectors of randomly cropped images, $\{\boldsymbol{z}_n'^I\}_{n=1}^m$, as described in Section 3.1. The contrastive loss function is effective for embedding contextually irrelevant data that are far apart in the multimodal embedding space.

Furthermore, to facilitate the training of the encoders to learn aesthetic image compositions, we design a hinge loss function $\mathcal{L}_H$ inspired by the margin ranking loss [27, 28] as follows:

$$\mathcal{L}_H = \sum_{i=1}^{m}\max\{0, \alpha - \mathrm{sim}(\boldsymbol{z}_i^A, \boldsymbol{z}_i^I) + \mathrm{sim}(\boldsymbol{z}_i^A, \boldsymbol{z}_i'^I)\} + \sum_{i=1}^{m}\max\{0, \alpha - \mathrm{sim}(\boldsymbol{z}_i^T, \boldsymbol{z}_i^I) + \mathrm{sim}(\boldsymbol{z}_i^T, \boldsymbol{z}_i'^I)\}, \quad (2)$$

where α is a margin. The hinge loss function is effective for embedding the original image closer to the feature vector of the corresponding music audio (or text) than that of the randomly cropped images in the space.

To leverage the advantages of both the contrastive and hinge loss functions, we formulate a novel loss function that incorporates those loss functions as follows:

$$\mathcal{L} = \mathcal{L}_C + \lambda_H \mathcal{L}_H, \quad (3)$$

where λ_H is a weight.

3.3 Cover Image Composition Method

We use the grid-anchor formulation [14] to generate image composition candidates with a specified aspect ratio from each image in the image collection. First, image composition candidates are automatically determined by using the grid-anchor formulation that constructs an image grid with $M \times N$ bins on the original image. Each bin has a width of Δw px and a height of Δh px. Then, the grid-anchor formulation chooses a region to be cropped by selecting two anchor points so that the region is at least 40 percent of the size of the original image. This formulation can efficiently determine image composition candidates of a specified aspect ratio from the original image. We use $\Delta w = \Delta h = 12$ when cropping an image with an aspect ratio of 1:1 and $\Delta w = 16, \Delta h = 9$ for 16:9. The values of M and N are automatically determined from the size of each bin and the aspect ratio of the original image. As a result, we can obtain a dozen to several hundred image composition candidates of a specified aspect ratio from each image in the image collection. Note that the number of image composition candidates generated by the proposed method depends on the size of each image.

Then, we can calculate how an image composition candidate matches a query as follows:

$$\mathcal{F}_i = \mathrm{sim}(\boldsymbol{z}^Q, \boldsymbol{z}_i^I) \tag{4}$$

where $\mathbf{z}^{\mathrm{Q}}$ and $\mathbf{z}_i^{\mathrm{I}}$ are the feature vectors of the query and the i-th instance of the image composition candidates, respectively. We calculate feature vectors of the query and image composition candidates using the trained encoders. By sorting the values $\{\mathcal{F}_i\}$ in descending order, we can obtain a ranked list of the image composition candidates.

4. EXPERIMENTS AND RESULTS

This section describes a qualitative analysis and comparison experiments to evaluate our method's effectiveness.

4.1 Dataset

We utilized three datasets (one image dataset and two multimodal datasets) in our experiments.

4.1.1 Image dataset

The **TAD66K dataset** [15] is specifically designed for image aesthetics assessment. The dataset contains over 66K images of various aspect ratios, covering 47 popular themes. We utilized this dataset as the image collection.

4.1.2 Multimodal dataset

The **YT8M-MusicVideo dataset** [16] is a subset of the YouTube-8M dataset [30], comprising videos tagged as "music video." We collected 73,113 triplets consisting of music audio (average length of 4 min with a 48 kHz sampling rate), the corresponding thumbnail image (an RGB image with an aspect ratio of 16:9), and the corresponding metadata including title, channel name, and upload date on YouTube from 60,785 YouTube channels. We randomly split the dataset into training (64,001 songs), validation (7,112 songs), and test (2,000 songs) sets with no channels overlapping across these sets.

The **AS5M dataset** is a private dataset containing triplets of a music audio excerpt (a 30 s audio preview for trial listening, with a 44.1 kHz sampling rate), the corresponding cover image (a square RGB image), and the corresponding metadata including song title, artist name, collection name, music genre, and release date. The dataset contains 5,920,828 music audio excerpts and their metadata by 174,629 artists, and 1,115,668 cover images. Because multiple excerpts from an album are associated with a single cover image, each image corresponds to about 5.3 excerpts on average. The excerpts, typically representative music sections, were already cropped on a music streaming service from which they were crawled. The corresponding cover images and metadata were crawled simultaneously. The songs encompass a variety of music genres (over 250, according to the streaming service). We randomly split the dataset into training, validation, and test sets with an eight-one-one ratio and with no artists or images overlapping across these sets. For evaluation, we constructed ten folds of test subsets by randomly selecting 2,000 triplets of a music audio excerpt, cover image, and text prompt for each fold from the test set.

We determined the size of each test set by following the setup used in related works [16, 31].

4.2 Qualitative Analysis

We performed a qualitative analysis to demonstrate that the proposed method can generate aesthetic image composition candidates and retrieve appropriate image compositions in response to each query. The most effective way to demonstrate the proposed method is to test it on real examples [16]. We thus used music audio available on YouTube [1] as music audio queries, and text available on Wikipedia [2] as text queries. The queries were selected on the basis of their high popularity, that is, hit charts (rankings) of music sales and YouTube views. For the image collection, we used all images in the TAD66K dataset [15].

Figure 3 shows example results of our method. The images in the top and bottom four rows are formatted in an aspect ratio of 16:9 and 1:1, respectively. From the ranked list, we list the top two images as "most matched", the middle three images as "moderately matched", and the bottom two as "least matched." The results show that our method captures the property of each query and retrieves image compositions that match either the music audio or text queries. For example, in the case of "White Christmas" (third row of Figure 3), the word "Christmas" resulted in the most matched images, such as the board shaped like Santa Claus, decorated Christmas tree, and star-shaped ornament. In the case of "Call Me Maybe" (fifth row of Figure 3), the top ranked results feature charming images since the song is categorized as teen-pop. Altogether,

[1] Each music audio query in Figure 3 can be accessed at https://youtu.be/{ID}.

[2] We used the first paragraph that introduces a song. Each text query in Figure 3 can be accessed at https://en.wikipedia.org/wiki/{Url}.

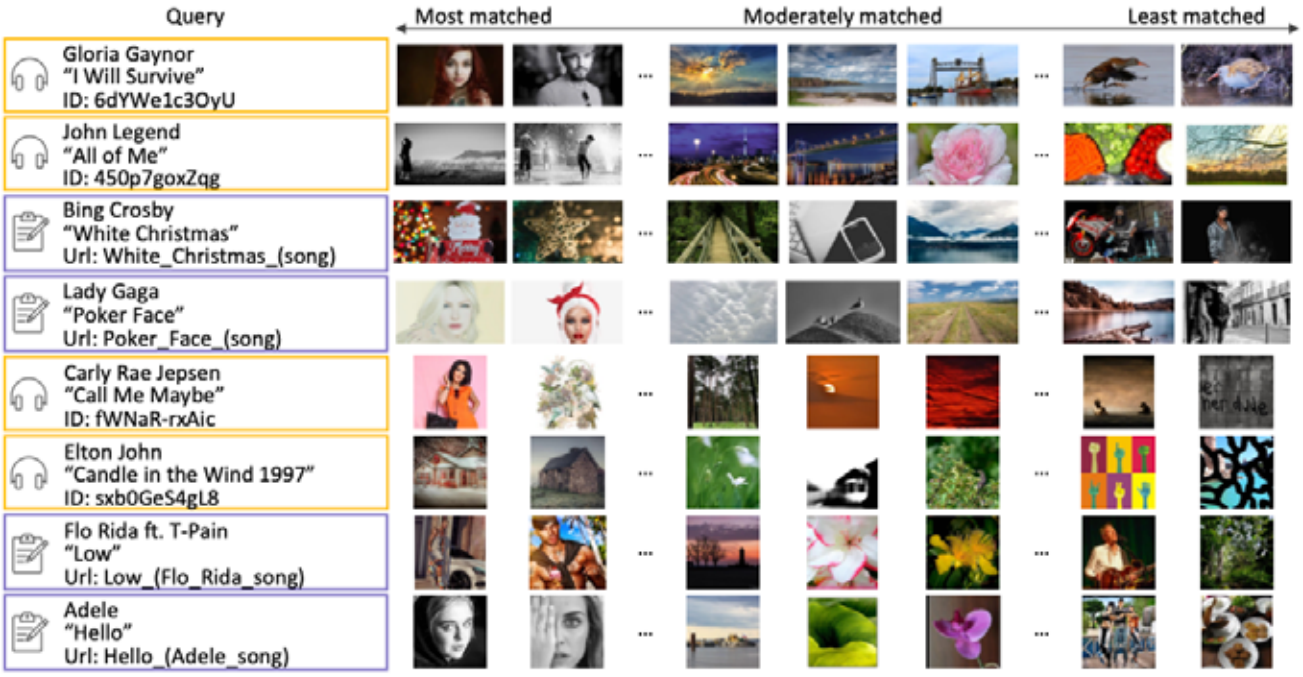


Figure 3. Example results, showcasing ranked lists of image composition candidates generated from the TAD66K dataset [29]. The query with the headphone icon represents a music audio query (https://youtu.be/6dYWe1c3OyU, etc.), while the query with the text icon represents a text query (https://en.wikipedia.org/wiki/White_Christmas_(song), etc.).

"most matched" images are similar in context, showing that the multimodal embedding space is well constructed. These results demonstrate that the proposed method can retrieve appropriate image compositions in response to each query.

4.3 Study on Multimodal Image Composition Learning

While training the encoders, we assumed that a combination of music audio, an image, and text from the same song is positive, whereas a combination from different songs is negative. In addition, we designed our loss function so that the aesthetic (i.e., original) image is closer to the corresponding music audio (or text) than the randomly cropped images in the multimodal embedding space. We thus quantitatively evaluated how closely the feature vectors of an original pair are located in the multimodal embedding space.

4.3.1 Experimental Setup

Training Details. Our implementation was based on PyTorch [32]. We used 16 NVIDIA V100 GPUs under each experimental condition. Each GPU computed 128 triplets of music audio, images, and text per iteration. When training the encoders, we used the Adam optimizer [33] with an initial learning rate of 1.0e-4. We set the weight λ_H to 1.0 and the margin α to 0.0. Following the setup in MoCo [11], we also set the temperature-scaling value τ to 0.07.

Evaluation Metrics. We used the recall@k (R@k), which is the standard evaluation metric in retrieval tasks, to assess the accuracy with which each method could find the original image corresponding to a music audio (or text) query from the image collection. R@k evaluates how much correct content is retrieved in the top results. A higher recall at a given k means that the retrieval method is more practical. We displayed R@k as a percentage.

In addition, we used the intersection over union (IoU), which is the standard evaluation metric in image cropping, to assess how precisely each method could identify the region of the original image within the outpainted image. For this evaluation, we used the original image and a hundred images randomly cropped from the outpainted images as the image composition candidates. We calculated the IoU between the region of the original image and that of the top ranked image composition retrieved by each method. We then averaged the obtained IoU scores over the test set.

4.3.2 Conditions

To demonstrate the effectiveness of our method, we compared the following four methods.

- **Baseline** solely used the contrastive loss function, where the similarity matrices only consist of cross-modal feature vectors (e.g., $\mathrm{sim}(\boldsymbol{z}_i^A, \boldsymbol{z}_j^I)$).
- **Baseline + Self-modal supervision** solely used the contrastive loss function, where the similarity matrices consist of self-modal feature vectors (e.g., $\mathrm{sim}(\boldsymbol{z}_i^A, \boldsymbol{z}_j^A)$) in addition to cross-modal feature vectors.
- **Baseline + Self-modal supervision + Outpainted images** solely used the contrastive loss function, where the similarity matrices consist of the feature vectors of randomly cropped images (e.g.,

Table 1. Results for R@k and IoU on the test set of the YT8M-MusicVideo dataset [16] with k set to 1, 5, and 10.

Method	Music-to-Image				Text-to-Image			
	R@1 ↑	R@5 ↑	R@10 ↑	IoU ↑	R@1 ↑	R@5 ↑	R@10 ↑	IoU ↑
Baseline	0.75	3.25	5.45	0.3415	3.45	8.10	10.60	0.4787
+ Self-modal supervision	0.75	3.35	5.95	0.3669	**3.70**	**8.15**	**10.75**	0.4872
+ Outpainted images	**0.85**	4.10	6.80	0.3722	3.60	7.95	10.60	0.5096
+ Hinge loss function (ours)	**0.85**	**4.30**	**7.15**	**0.4415**	3.55	8.00	**10.75**	**0.5271**

Table 2. Results for R@k and IoU on the test subsets of the AS5M dataset with k set to 1.

Method	Music-to-Image		Text-to-Image	
	R@1 ↑	IoU ↑	R@1 ↑	IoU ↑
Baseline	1.77 ± 0.27	0.4294 ± 0.009	6.57 ± 0.33	0.6679 ± 0.005
+ Self-modal supervision	2.22 ± 0.18	0.4524 ± 0.006	6.40 ± 0.34	0.6754 ± 0.006
+ Outpainted images	2.25 ± 0.26	0.4391 ± 0.005	6.39 ± 0.36	$\mathbf{0.6880 \pm 0.007}$
+ Hinge loss function (ours)	$\mathbf{2.44 \pm 0.25}$	$\mathbf{0.4766 \pm 0.008}$	$\mathbf{6.70 \pm 0.31}$	0.6817 ± 0.007

$\mathrm{sim}(z_i^I, z_j'^I)$) in addition to self- and cross-modal feature vectors.

- **Baseline + Self-modal supervision + Outpainted images + Hinge loss function**, which is our proposed method, used both the contrastive and hinge loss functions as described in Section 3.2.

To quantitatively evaluate the performance of each method, we set up two tasks, Music-to-Image and Text-to-Image, in which music audio and text were used as a query, respectively, to retrieve a corresponding image and identify a corresponding region.

4.3.3 Results

Table 1 shows that our method generally outperformed the baseline method by 0.1 points for R@1, 1.05 points for R@5, 1.7 points for R@10, and 0.1 points for IoU in the Music-to-Image task, and by 0.1 points for R@1, 0.15 points for R@10, and 0.0484 points for IoU in the Text-to-Image task on the YT8M-MusicVideo dataset. Likewise, Table 2 shows that our method outperformed the baseline method by 0.67 points for R@1 and 0.0472 points for IoU in the Music-to-Image task, and by 0.13 points for R@1 and 0.0138 points for IoU in the Text-to-Image task on the AS5M dataset.

These results demonstrate the effectiveness of our multimodal image composition learning. That is, our method can closely align the feature vector of the given query with that of the corresponding image of high aesthetic quality in the multimodal embedding space. Thus, our proposed method can effectively retrieve image compositions that are suitable for the query.

4.3.4 Visualization of Feature Vectors

We further investigated the nature of obtained feature vectors to clarify which query properties facilitate our proposed multimodal image composition learning. For example, Libeks et al. found that cover images were characterized by music genre [35]. Here, we explored music genre and artist tags. We used the test subsets of the AS5M dataset. The feature vectors were visualized by using t-SNE [34] to project them to a two-dimensional space.

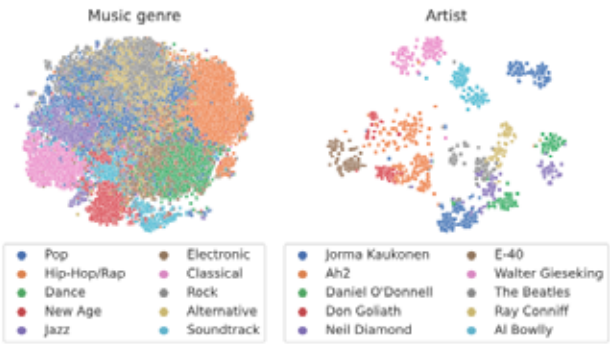


Figure 4. Visualization of music audio, image, and text feature vectors in the test subsets of the AS5M dataset with respect to music genre tags (left) and artist tags (right). These are visualized using t-SNE [34].

The left side of Figure 4 visualizes feature vectors by genre tags. We used the feature vectors of the music audio, images, and text for 12,180 songs in the top 10 most popular genres in the dataset. The result shows that feature vectors form clusters for each of several genres, such as Hip-Hop/Rap and Classical. That is, music audio, images, and text in each of these genres are closely associated with each other. The right side of Figure 4 visualizes feature vectors by artist tags. We used the feature vectors for 757 songs by the top 10 most popular artists in the dataset. Surprisingly, the result shows that feature vectors are clearly separated by each artist tag, while we simply assumed that a combination of music audio, an image, and text from the same song is positive. That is, music embodies the unique personality of each artist.

These results demonstrate that our proposed method successfully trains the encoders so that the feature vectors of music audio, images, and text with similar contexts are

close to each other in the multimodal embedding space.

5. DISCUSSION

The proposed method is useful in various situations, e.g., when musicians want to add a cover (or thumbnail) image to their own music, using a large collection of images or photographs that they personally own or that are free of rights issues. The retrieved image composition can be used as is or further adjusted as a cover (or thumbnail) image.

In such a situation, one might wonder why not use recent multimodal generative models capable of generating cover images from various inputs [36]. A primary concern with generative techniques is the potential to replicate existing images from training data [37, 38]. For musicians, legal ambiguities and risks of using generated images can be a substantial drawback.

In contrast, the use of self-collected or owned images or commercial stock images is free of rights concerns and is likely to be the preferred option, especially for commercial use. Therefore, this paper focuses on the unique challenge of retrieving cover image compositions from an image collection without daring to consider using the generative models.

6. CONCLUSION

We proposed a method of retrieving image compositions suitable for a music audio (or text) query. The contributions of this paper can be summarized as follows. First, to ensure that image compositions have the specified aspect ratio, high aesthetic quality, and contextual relevance to the query, we utilized multimodal image retrieval and cropping techniques, which offer a novel solution to this challenge. Second, we demonstrated that the proposed method can retrieve image compositions suitable for queries, as our qualitative analysis shows. Third, our proposed method succeeded in training the encoders so that the feature vectors of music audio, images, and text from the same song (i.e., those with similar contexts) are close to each other in the multimodal embedding space, as demonstrated by our quantitative evaluation.

This work will lead to further development of assistive tools that can deal with various music content.

Acknowledgments

This work was supported in part by JST CREST Grant Number JPMJCR20D4 and JSPS KAKENHI Grant Number 22K18017, Japan.

7. REFERENCES

[1] M. Vad, "The album cover," *J. Pop. Music Stud.*, vol. 33, no. 3, pp. 11–15, 2021.

[2] S. J. Cunningham and D. M. Nichols, "How people find videos," in *Proceedings of the ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, 2008, pp. 201–210.

[3] A. Inc., "Adobe stock," 2023, retrieved March 1, 2024 from https://stock.adobe.com.

[4] I. Getty Images Holdings, "Unsplash," 2023, retrieved March 1, 2024 from https://unsplash.com.

[5] Y. Deng, C. C. Loy, and X. Tang, "Image aesthetic assessment: An experimental survey," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 80–106, 2017.

[6] A. Santella, M. Agrawala, D. DeCarlo, D. Salesin, and M. Cohen, "Gaze-based interaction for semi-automatic photo cropping," in *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI)*, 2006, pp. 771–780.

[7] S. Bhattacharya, R. Sukthankar, and M. Shah, "A framework for photo-quality assessment and enhancement based on visual aesthetics," in *Proceedings of the ACM International Conference on Multimedia (MM)*, 2010, pp. 271–280.

[8] N. Horanyi, K. Xia, K. M. Yi, A. K. Bojja, A. Leonardis, and H. J. Chang, "Repurposing existing deep networks for caption and aesthetic-guided image cropping," *Pattern Recognit.*, vol. 126, p. 108485, 2022.

[9] Z. Zhong, M. Cheng, Z. Wu, Y. Yuan, Y. Zheng, J. Li, H. Hu, S. Lin, Y. Sato, and I. Sato, "ClipCrop: Conditioned cropping driven by vision-language model," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023, pp. 294–304.

[10] K. Sohn, "Improved deep metric learning with multi-class n-pair loss objective," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS)*, 2016, pp. 1857–1865.

[11] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[12] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 9729–9738.

[13] W. Li, Z. Lin, K. Zhou, L. Qi, Y. Wang, and J. Jia, "MAT: Mask-aware transformer for large hole image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 10 758–10 768.

[14] H. Zeng, L. Li, Z. Cao, and L. Zhang, "Grid anchor based image cropping: A new benchmark and an efficient model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1304–1319, 2022.

[15] G. Jia, H. Huang, C. Fu, and R. He, "Rethinking image cropping: Exploring diverse compositions from global views," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 2446–2455.

[16] D. Surís, C. Vondrick, B. Russell, and J. Salamon, "It's time for artistic correspondence in music and video," in *Proceedings of the IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*, 2022, pp. 10 564–10 574.

[17] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs, "Automatic thumbnail cropping and its effectiveness," in *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*, 2003, pp. 95–104.

[18] J. Zhang, Y. Miao, and J. Yu, "A comprehensive survey on computational aesthetic evaluation of visual art images: Metrics and challenges," *IEEE Access*, vol. 9, pp. 77 164–77 187, 2021.

[19] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, 2020, pp. 213–229.

[20] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[21] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. L. Scao, S. Gugger, M. Drame, Q. Lhoest, and A. M. Rush, "Transformers: State-of-the-art natural language processing," in *Proceedings of the Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP-SD)*, 2020, pp. 38–45.

[22] Y. Luo and N. Mesgarani, "Conv-TasNet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 27, no. 8, pp. 1256–1266, 2019.

[23] R. Takahashi, T. Matsubara, and K. Uehara, "Data augmentation using random image cropping and patching for deep CNNs," *IEEE Trans. Circuits. Syst. Video Technol.*, vol. 30, no. 9, pp. 2917–2931, 2019.

[24] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020, pp. 13 001–13 008.

[25] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proceedings of the International Conference on Machine Learning (ICML)*, 2021, pp. 8748–8763.

[26] T. Sellam, D. Das, and A. P. Parikh, "BLEURT: Learning robust metrics for text generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2020, pp. 7881–7892.

[27] F. Schroff, D. Kalenichenko, and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 815–823.

[28] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu, "Learning fine-grained image similarity with deep ranking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 1386–1393.

[29] S. He, Y. Zhang, R. Xie, D. Jiang, and A. Ming, "Rethinking image aesthetics assessment: Models, datasets and benchmarks," in *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2022, pp. 942–948.

[30] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "YouTube-8M: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[31] L. Prétet, G. Richard, and G. Peeters, "Cross-modal music-video recommendation: A study of design choices," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2021, pp. 1–9.

[32] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, 2019, pp. 8024–8035.

[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015, pp. 1–13.

[34] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2605, 2008.

[35] J. Libeks and D. Turnbull, "You can judge an artist by an album cover: Using images for music annotation," *IEEE Trans. MultiMedia*, vol. 18, no. 4, pp. 30–37, 2011.

[36] S. Wu, H. Fei, L. Qu, W. Ji, and T.-S. Chua, "NExT-GPT: Any-to-any multimodal LLM," *arXiv preprint arXiv:2309.05519*, 2023.

[37] G. Somepalli, V. Singla, M. Goldblum, J. Geiping, and T. Goldstein, "Diffusion art or digital forgery? investigating data replication in diffusion models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 6048–6058.

[38] ——, "Understanding and mitigating copying in diffusion models," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 36, pp. 47 783–47 803, 2023.

PROGRAM NOTES

CONCERT 2

/ Wood, metal and tension - for violin and electronics

/ Cryoconite

/ Mirrored: ceilings, floors, walls

/ TransVariations – Music beyond the limits of time and technology

WOOD, METAL AND TENSION - FOR VIOLIN AND ELECTRONICS

Cárthach Ó Nuanáin
Cork School of Music

This piece continues a cycle of compositions known as Materials that explores the interaction between acoustic instruments and real-time analysis and intelligent organisation of timbres using concatenative synthesis. In a process that is antithesis and antagonistic of Big Data, I am fascinated by the seemingly endless sound shapes and sonic events that can be squeezed out of smaller, intimate collections of carefully-curated sounds through selection and myriad post transformations that render the original unintelligible.

The piece is built around the Deacon [https://github.com/carthach/deacon] instrument where collections of sound samples are loaded into a Max/MSP device that segment the samples by onset times then analyse their spectral, timbral and perceptual characteristics. The device can then listen to live input and select and play the closest matched samples using those timbral fingerprints.

The instrument was conceived for my acousmatic piece Sem Cordas, selected as a highlight work of the International Computer Music Conference in 2021 [https://artes.uc.cl/noticias/concierto-presenta-las-mejores-obras-musicales-en-formato-acusmatico-y-audiovisual-de-la-icmc–2021/]. I first experimented by recording long improvisations with a detuned Irish tenor banjo, exploiting the sharp attacks to create curious collisions of sounds with the computer. Hours of material was then edited down to form the cohesive piece I was satisfied with.

Since then I have elaborated instrument and technique to enable live performance with other instrumentalists. This brought with it significant challenges. The instrument needed to be refined to be less chaotic, more controlled and efficient in its role as automaton accompanist. A notation system was devised using MuseScore and Inkcape to engrave graphic scores to help human performers navigate a piece. This culminated in metal and air for flute and electronics - first performed in 2022 [https://soundcloud.com/carthach/metal-and-air-flute-and-electronics].

wood, metal and tension for solo violin and electronics renews a focus on mechanical and digital materiality. The corpus of sound curated is thematically related to the physical qualities of the violin, and includes recordings of wooden string- based instruments like the bouzouki and fiddle along with metallic impacts, scrapes and other electronic noises from sampled sources and synthesisers.

The piece is divided into 4 sections according to the electronic material that responds to the violin player's performance.

Section A is mostly percussive, noisy electronics with corresponding extended techniques for the violin. Section B contains more coherent pitched material for both electronics and violin. Section C combines material and approaches from A and. Section D concludes with dissipating harmonics coupled with bouzouki samples.

Sound spatialisation of the electronic sources is achieved using the E4L open-source tools for Ambisonics 3D panning [McArthur, Angela, et al. "A survey of 3D audio through the browser: practitioner perspectives." 2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA). IEEE, 2021.]. This allows flexible deferral of final speaker configuration to the point of performance.

Vitor Vieira (ESMAE), violin

CRYOCONITE

Enrico Dorigatti
University of Portsmouth

Cryoconite is inspired by the homonymous phenomenon that sees snow and ice melting faster due to the windblown dust covering portions of the frozen layer. As a result, the surface gets covered by holes—hence the name. The piece structure reflects such phenomenon as, over the whole duration, sonic events of different lengths (the holes) superimpose over more static layers (the virgin frozen surface). For its composition, Cryoconite relies on found sounds, whilst, concerning sound design techniques, the material has undergone several processing iterations, transforming it beyond recognition.

MIRRORED: CEILINGS, FLOORS, WALLS

youngjae cho

I concretize what I experience in everyday life, for example a night sky with different star constellations or countless coloured books on the bookshelf. In this piece, I also imagined a room where the walls are completely covered with mirrors. These many multidimensional mirrors have a fascinating effect. The images, which change depending on the vantage point, seem to relate to the resonance and the sonic environment. This work is intended to be presented in a dome of loudspeakers version (3D) or any multi channel version (16 ch or 8ch) in the concert hall.

TRANSVARIATIONS – MUSIC BEYOND THE LIMITS OF TIME AND TECHNOLOGY

Oeyvind Brandtsegg
Norwegian University of Science and Technology

Alfonso Benetti
University of Aveiro

Trond Engum
Norwegian University of Science and Technology

Francisco Monteiro
Instituto Politécnico do Porto

The project uses innovative technological approaches to problematize the relationship between artistic creation and interpretation in the context of Western-Art Music. It uses a well-known classical composition as a departure point and research object, namely Beethoven's "32 Variations in C minor". We ask: How would this piece sound if the composer had all modern tools of today at his disposal? Well aware that the question itself has a logical flaw (as the cultural implications would surely mean that the whole piece would be different), we investigate the relationship between score and sonic manipulation of the instrument. This again leads to distinctive changes in the performer's phrasing and articulation, and as such can radically change the basis on which interpretation is made. Keeping in mind that the instrument for which the piece was composed was new technology at the time, it seems natural to make use of today's technology in a contemporary interpretation. With the full freedom of modern sonic manipulation, adhering to the tradional score might seem limiting, as the affordances of the new instrumental sound also affect which notes to play. This can also be seen as a point of friction and dissonance, which demands and generates creave soluotins.

The project is a collaboraon between Alfonso Benetti (pianist), Øyvind Brandtsegg (live sound manipulation and programming), Trond Engum (live sound manipulation and recording), and Francisco Monteiro (external ear). Three academic instutions participate in the project: University of Aveiro (Benetti), Instituto Politécnico do Porto (Monteiro), and Norwegian University of Science and Technology (Brandtsegg and Engum).

INSTALLATION

PROGRAM NOTES

INS-
TALLA-
TION

TRANSIMMANENCY - SYMBIOTIC TREATMENTS, TRANS-INTERACTIVE INSTALLATION WITH WEB AND BRIGHT RESONANT OBJECTS

Lorenzo Ballerini
Conservatory of Trapani

Alberto Gatti
Conservatorio di Musica Giacomo Puccini

Giuseppe Ernandez
Conservatory of Trapani

Massimo Reina
Conservatory of Trapani

"Transimmanency - Symbiotic Treatments" is a trans- interactive installation that, through web technology and bright resonant objects, reshapes human immanence by immersing it in a new audiovisual realm. The aim of the installation is to offer visitors the opportunity to challenge the society of control through the same mechanisms that govern it, providing them a medium to express their creativity and celebrate their uniqueness. Through the use of vibrating transducers, light, and customized web interactions, visitor's audiovisual generated content is transformed into a new sensory experience, transcending predefined rules and established power structures.

10:00–12:00 **Paper & Poster/Demo session 6**

Introducing MakamNetz – A Virtual Guide to Turkish Makam Universe

Ozan Baysal (Istanbul Technical University)
Recep Gül (Istanbul Technical University)
Yusuf Can Şeftali (Istanbul Technical University)
Gümrah Sindar (Istanbul Technical University)
Zülfü Yalçın (Istanbul Technical University)
Gözde Çolakoğlu Sarı (Istanbul Technical University)

Module-level MIDI

Sean Luke (George Mason University)
John Tuffen (wonkystuff)
Mathias Brüssel (tangible waves)

MIDI 1.5, So to Speak

Sean Luke (George Mason University)
Luca Ludovico (Università degli Studi di Milano)

libremidi: a cross-platform library for real-time MIDI 1 and 2

Jean-Michaël Celerier (Société des Arts Technologiques)

Programming FPGA Platforms for Real-Time Audio Signal Processing in C++

Pierre Cochard (Inria)
Maxime Popoff (INSA Lyon)
Romain Michon (Inria)
Tanguy Risset (Insa-Lyon/Inria)

Enabling Affordable and Scalable Audio Spatialization With Multichannel Audio Expansion Boards for FPGA

Maxime Popoff (INSA Lyon)
Romain Michon (Inria)
Tanguy Risset (Insa-Lyon/Inria)

Perspectives on Immersion Through Laser Doppler Vibrometry

Roderick Buchanan-Dunlop (University of Edinburgh)
Michael Newton (University of Edinburgh)
Martin Parker (University of Edinburgh)

Conditioning Methods for Neural Audio Effects

Riccardo Simionato (Universirty of Oslo)
Stefano Fasciani (University of Oslo)

12:00 **Concert session 3**

les espaces negatifs

Pierre Alexandre Tremblay (University of Huddersfield)

The Voices

Ji Youn Kang (IEM, KUG Graz)

'Rust' for piano with subtle electronics

Luís Neto da Costa (Aristotle University of Thessaloniki)

14:30 Keynote Speaker

Sound from everywhere: Immersive Realities and Fictions in Sound-art and Composition
Natasha Barrett

15:30 - 16:30 **Paper & Poster/Demo session 7**

Hybrid Neural Audio Effects

Riccardo Simionato (Universirty of Oslo)
Stefano Fasciani (University of Oslo)

Implementation of a Continuously Variable Delay Line by Crossfading Between Several Tap Delays

Thibaut Carpentier (Ircam)

Real-time Psychoacoustic Frequency Masking Compensation for Audio Signals with Overlapping Spectra

Giorgio Presti (Università degli Studi di Milano)
Nicola Degiorgi (Politecnico di Torino)
Amedeo Fresia (Politecnico di Torino)
Antonio Servetti (Politecnico di Torino)

Modeling the Pultec EQP-1A with Wave Digital Filters

Alberto Barrera (Universitat Pompeu Fabra)
Xavier Lizarraga-Seijas (Music Technology Group - Universitat Pompeu Fabra)
Frederic Font (Music Technology Group - Universitat Pompeu Fabra)
Revitalizing Multimedia Art: Restoration and Re-Activation of an Audiovisual Installation in Virtual Reality

Alessandro Russo (University of Padua)
Andrea Franceschini (Università di Padova)
Antonio Rodà (University of Padova)
Sergio Canazza (University of Padova)

Music Theatre and Digital Archives

Ana Filipa Gonçalves de Magalhães (CESEM NOVA FCSH)

Phase Repair for Time-Domain Convolutional Neural Networks in Music Super-Resolution

Yenan Zhang (Waseda University)
Guilly Kolkman (University of Amsterdam,)
Hiroshi Watanabe (Waseda University)

Flow: a Mutitimbral, Polyphonic, Modular, Additive Music Synthesizer

Sean Luke (George Mason University)

Loom: a Speculation of Oral Tradition in Immersive Media

Yingjia Guo (Stanford University)
Mengtai Zhang (Fluffy Cactus)

Exploring the Space Between I nstrument and Controller

Eleonora Oreggia (Goldsmiths)

17:30–19:30 **Paper & Poster/Demo session 8**

Balancing Physical Modeling and Musical Requirements: Algorithmically Simulating the Calls of Hyalessa Maculaticollis for Real-time Instrumental Control

Staas de Jong (Universiteit Leiden)

A Universal Tool for Generating Datasets from Audio Effects

Stefano Fasciani (University of Oslo)
Riccardo Simionato (Universirty of Oslo)
Aleksander Tidemann (University of Oslo)

Stochastic Resonance: Molding Sounds from Noise

Esteban A Gutiérrez (Pontificia Universidad Católica de Chile)
Rodrigo F Cadiz (Pontificia Universidad Catolica de Chile)

Speculative Machine Learning in Sound Synthesis

Luc Döbereiner (IEM, KUG Graz)
David Pirrò (IEM, KUG Graz)

Unveiling the Timbre Landscape: a Layered Analysis of Tenor Saxophone in RAVE Models

Nádia Carvalho (University of Porto, Faculty of Engineering & INESC TEC)
Jorge Sousa (University of Aveiro & INET-md)
Gilberto Bernardes (INESC TEC & University of Porto, Faculty of Engineering)
Henrique Portovedo (University of Aveiro)

Tempo Estimation on Multiple Metrical Levels with Sequency Flux and Multiresolution Analysis

Savvas Kazazis (Aristotle University of Thessaloniki)

Correlations Between Objective and Subjective Evaluations of Music Source Separation

Erika Rumbold (University of Victoria)
George Tzanetakis (University of Victoria)
Bryan Pardo (Northwestern University)

Reconstructing the Charlie Parker Omnibook Using an Audio-to-score Automatic Transcription Pipeline

Xavier Riley (C4DM)
Simon Dixon (Queen Mary University of London)

Exploring Deep Learning Methodologies for Music Emotion Recognition

Pedro L. Louro (DEI - FCTUC, University of Coimbra)
Hugo Redinho (CISUC)
Ricardo S Malheiro (CISUC / Polytechnic Institute of Leiria)
Rui P Paiva (DEI - FCTUC, University of Coimbra)
Renato Panda (Ci2.ipt, CISUC)

ChordSync: Conformer-Based Alignment of Chord Annotations to Music Audio

Andrea Poltronieri (University of Bologna)
Valentina Presutti (University of Bologna)
Martín Rocamora (Universitat Pompeu Fabra)

Prevalence Of Tresillo Rhythm In Contemporary Popular Music

Pushkar Jajoria (University of Galway)
Aurel Ruben Mäder (EPFL)
James McDermott (University of Galway)

19:30 **Concert 4**

saccades

Ted Moore
Jorge Sousa (University of Aveiro, INET-md)

nan

Ted Moore

kurivari-live

Alo Allik (tehis.net)

KEY NOTE

SOUND FROM EVERYWHERE: IMMERSIVE REALITIES AND FICTIONS IN SOUND-ART AND COMPOSITION

NATASHA BARRETT

DAY 3

SOUND FROM EVERYWHERE:

Immersive Realities and Fictions in Sound-art and Composition

NATASHA BARRETT

When I was invited to present a keynote at SMC2024, I was inspired by the conference's central question: what makes an experience truly immersive? This concept can be interpreted in various ways, influenced by the listener, the context, and the technology available. Even the definition of 'immersive' is open to interpretation.

I decided to shape my presentation around these considerations, focusing on three immersive themes in my work with Ambisonics to create and control 3D soundfields in artistic interventions. These themes include site-specific outdoor immersive experiences in urban noise habitats, composed concert works, and live improvised immersive performances. A common theme across all these threads is the transformative role of the latest tools in realising our creative ideas.

This progression results in an accumulation of knowledge and creativity in my artistic practice, rather than merely replacing the old with the new. At the end of the presentation I will play a real-time demo, an extract from an outdoor immersive sound installation, and insights into my current work in progress "Toxic Colour".

NATASHA BARRETT

(NO/ UK) composes concert works, public space sound-art installations and multimedia interactive music using a broad palette of sounds, new technologies and experimental techniques.

She is widely known for her electroacoustic and acousmatic music, and use of 3D sound technology in composition. Her work is commissioned and performed throughout the world and has received over 20 international awards including the Nordic Council Music Prize, the Giga-Hertz Award (Germany), five prizes and the Euphonie D'Or in the Bourges International Electroacoustic Music Awards (France), two first prizes in the International Rostrum for electroacoustic music and most recently the honorary Thomas Seelig Fixed Media Award for 2023.

She regularly collaborates with performers, visual artists, architects and scientists, is active as a performer of live-electronics and spatial audio, as a researcher has a track record in both artistic and academic publications and has held professorships at institutions in Norway and Denmark.

PAPER & POSTER /DEMO SESSION 6

/ Introducing MakamNetz – A Virtual Guide to Turkish Makam Universe

/ Module-level MIDI

/ MIDI 1.5, So to Speak

/ Libremidi: a cross-platform library for real-time MIDI 1 and 2

/ Programming FPGA Platforms for Real-Time Audio Signal Processing in C++

/ Enabling Affordable and Scalable Audio Spatialization With Multichannel Audio Expansion Boards for FPGA

/ Perspectives on Immersion Through Laser Doppler Vibrometry

/ Conditioning Methods for Neural Audio Effects

INTRODUCING MAKAMNETZ: A VIRTUAL GUIDE TO TURKISH MAKAM UNIVERSE

Ozan Baysal[1], Recep Gül[2], Yusuf Can Şeftali[3],Gümrah Sindar[4], Zülfü Yalçın[5], Gözde Çolakoğlu Sarı[6]
[1,2,6]Istanbul Technical University, Turkish Music State Conservatory
[3,4,5]Graduate School, Istanbul Technical University
{ozanbaysal, recepgul, seftali, sindar18, yalcinmuha, colakoglug@itu.edu.tr}

ABSTRACT

This presentation introduces MakamNetz, a virtual model crafted to map the musical universe of Turkish makam based on the relations of its nuclear substructures. Through interactive applications, MakamNetz serves as a virtual cartography tool, visually delineating harmonic connections among diverse makam structures, illuminating transitional pathways. For the theoretical background, emphasis is placed on the pivotal role of perde as the primary material of makam, and tetrachordal structures called çeşni (flavor), which act as modular building blocks of makam. In brief, our model interprets makam as progressions of Perde/Çeşni structures, highlighting the melodic path (seyir) of each makam. The model utilizes shared substructures and various transformational relationships, proposing three main categories: Intraconnected, Transference, and Fixed-axis, each with subtypes as well as combinations. MakamNetz offers a comprehensive understanding of these connections, displaying diverse Perde/Çeşni networks. The application's backend module is developed using Python and the Django framework, incorporaing transformational and relationship algorithms written in Python. Utilizing force-directed network graphs, the virtual model demonstrates different makam-spaces, guides through neighboring constellations, and allows hyperjumps to seemingly remote systems.

1. INTRODUCTION

Music-theory scholars have long sought to understand the harmonic connections within different musical structures. Visual models of pitch-set relationships have emerged as a useful tool in this pursuit, offering a panoramic perspective that allows for a holistic view of the musical universe. In particular, network models can reveal previously unrecognized pathways within the web of harmonious relations, offering a broader understanding of the underlying connections and patterns that govern musical composition.

One famous example of such a model is the tone-network "tonnetz" proposed by Leonhard Euler in the 18th century. This model later became the foundational framework for many theoreticians, including Hugo Riemann, who demonstrated his principles of harmonic motion and transformational models in the 19th century through the conceptual basis provided by the "tonnetz" [1, 2]. Similar representations can be found in the theoretical corpus of Turkish makam music, such as the manuscript treatise *Tefhimu¨'l Makâmât fi Tevlidi'n Nagâmât* ('Explanation of Makams in the Generation of Melodies') from the 18th century [3]. In this treatise, one can see the linking of different *perde* and makam structures as well as their interrelationships with other systems (Figure 1).

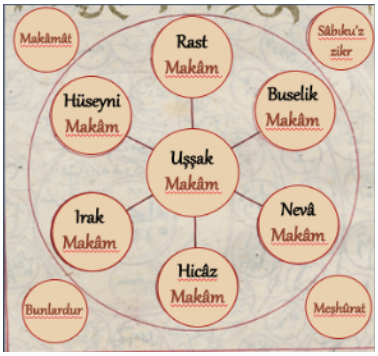


Figure 1. A reproduction of makam connections as depicted in *Tefhimü'l Makâmât* (Kemani Hızır Aga, 8A).

Recently, computer-aided tools have highlighted the gap between theory and practice in Turkish makam music. By utilizing advanced technologies such as tuning analysis, segmentation, statistical classification, and automatic transcription, researchers have developed sophisticated methods tailored to the unique elements of makam. [1]

This presentation offers a virtual model for mapping the makam universes of Turkish music according to their harmonic relations, that is, according to their common and/or related substructures. The network model presented here not only considers common substructures between different makams, but also recognize and include various relationships transformations and topographies between dif-

[1] Bozkurt (2008) [4] developed an automatic tuning analysis method, followed by Tamer and Bozkurt's (2013) [5] creation of microtonal tuners for custom tunings. Bozkurt (2014) [6] introduced automatic tonic detection and histogram-based analysis applications. The same year, Bozkurt et al. (2014a) [7] presented melodic segmentation tools using makam and usul features and published a comprehensive review of computational analysis in Turkish makam music (Bozkurt et al., 2014b) [8]. Benetos and Holzapfel (2015) [9] developed transcription systems for microtonal Turkish music, while Bozkurt (2015) [10] analyzed melodic progression in Turkish makam music. In 2016, Senturk (2016) [11] introduced Tomato, a Python toolbox for analyzing Turkish-Ottoman makam music, and Karakurt et al. (2016) [12] released MORTY, a mode recognition and tonic identification toolbox.

Copyright: © 2024 Ozan Baysal et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

ferent substructures. After reviewing some basic concepts and terminology, the first part of the presentation will explain the methodology of determining the relationships between various makam systems. Here we will present common nuclear elements and certain relationship principles that mutate such elements, creating alternative bridges with other makam systems.

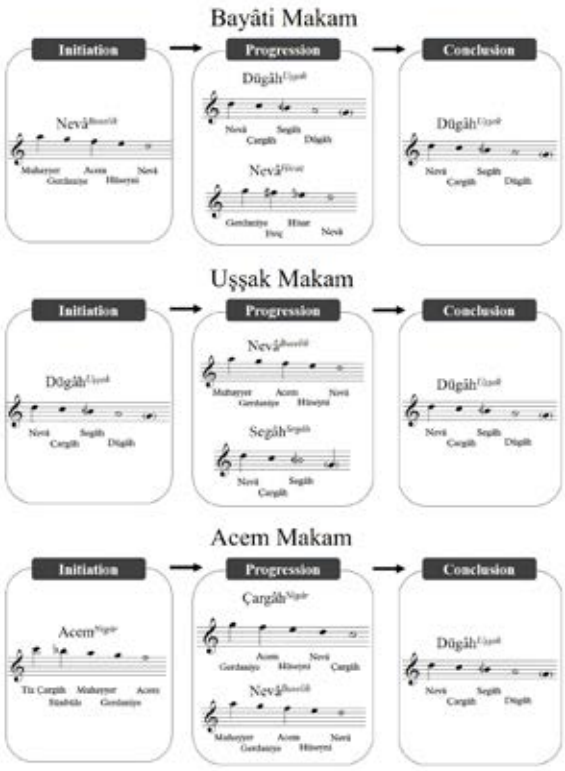


Figure 2. *Seyir* of *Bayâti*, *Uşşak* and *Acem* Makamlar as a chain of Perde^{Çeşni} progressions.

2. THEORETICAL BACKGROUND AND MODEL

The primary material of makam is *perde* (pl. *perdeler*), which in some cases can refer to a specific frequency – as in the Western notion of pitch -, or can encompass a range of frequencies that can vary according to the makam it is used in and its function in the melody. Thus a *perde* can be either stable or unstable (movable/dynamic) depending on the musical context. [2] Stable *perdeler* possess melodic-axis qualities, which function either as a finalis, a reciting-tone, or a suspended cadence; all of these can be used as anchoring pivots while connecting to other substructures. These modular building blocks of makam - similar to Greek genus/genera or Arabic jins/ajna - will be referred to as it is popularly used in Turkish as; *çeşni* - meaning flavor, expressing the temporary sonorous effect of the melody. Recently, Abu Shumays (2013) proposed a clear explanation of how Arabic ajna create a network of pathways that build up a maqam and the surrounding "jins baggage" they possess, an alternative/supplement to the conventional "scalar" explanation of maqamat [14]. The proposal holds true for Turkish makam music as well. We will be using Perde^{Çeşni} denotation to refer to specific *çeşni* structures built on specific perdeler. For instance, Neva^{Hicaz} means Neva *perde* with Hicaz *çeşni*.

A defining element of a makam is *seyir* (the course), the basic outline and the distinct melodic path of a makam. *Seyir* distinguishes makams that have the same *perde* structures and finalis degrees, but are different in their melodic orientation. We believe this is due to the combination of different Perde^{Çeşni} progressions. Thus, our model interprets makam as progressions of Perde^{Çeşni} structures, highlighting the melodic path (initiation, development, closure), or *seyir*, of each makam.

In Figure 2, the basic *seyir* of *Bayâti*, *Uşşak* and *Acem* makam are demonstrated in this manner. The whole notes represent the primary axis, the *perde* on which the *çeşni* is built on. The quarter notes are secondary axes, while the unstemmed full notes, although usually being the identifiers of the *çeşni*, are less significant in connecting the structural pillars due to their unstable inclinations [3] . Realize that all three makams presented in Figure 2 share two Perde^{Çeşni} structures, Neva^{Buselik} and Dügah^{Uşşak}. Using these common substructures as pivots, one can smoothly wander within the three listed makams. However, inter-makam transitions are not solely reliant on the utilization of such common Perde^{Çeşni} substructures. Indeed, each Perde^{Çeşni} may uniquely be connected to another, facilitating various types of transitions.

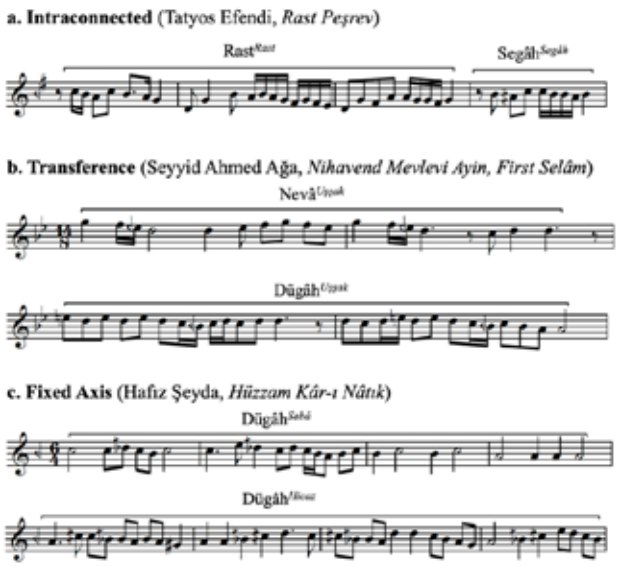


Figure 3. Transformation examples from the repertoir.

We propose three main Perde^{Çeşni} transformational relation categories;

(1)*Intraconnected*, in which the primary, the secondary, or both axes are displaced either within the same perde collection or its subsets/supersets;

[2] Recent computational research exposed this fact by various frequency-histogram analyses of makam performances, at which some perde regions are displayed as steep and narrow while some others resemble plateaus that spread over a bandwidth of nearly a semitone [13].

[3] For a discussion on the melodic inclination of different 'nuclei' substructure in makam music see Bayraktarkatal and Güray, 2023 [15]

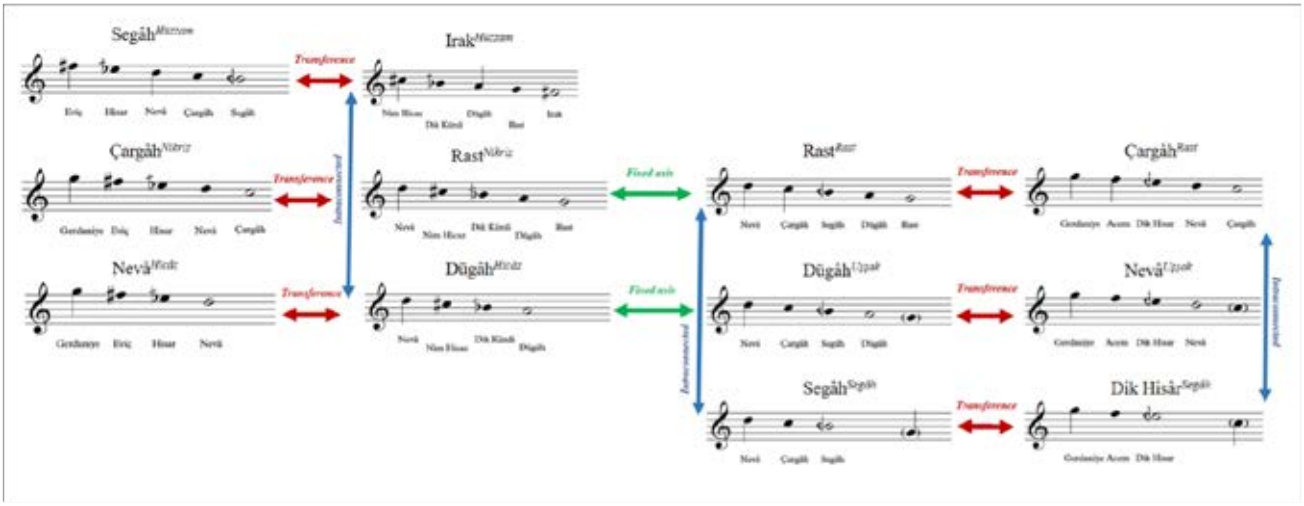


Figure 4. Three basic Perde^{Çeşni} transformations.

(2)*Transference*, in which the primary axis *perde* changes, but the *çeşni* structure remains the same (thus, the structure is basically transposed) in our model we limited the transferences with P5 and P4 (each up and down) as frequently witnessed in the repertoire;

(3)*Fixed axis*, in which one or two of the axis *perdeler* remain constant while the *çeşni* structures change.

One can come across many similar transformations and relations within the repertoire. Figure 3 provides three examples for each transformation type.

Figure 4 demonstrates examples of these three main transformations by connecting various Perde^{Çeşni} structures. Realize how Dügah^{Uşşak} is related to Neva^{Uşşak}and Dügah^{Hicaz} is related to Neva^{Hicaz} through transference, while Dügah^{Uşşak} and Dügah^{Hicaz} are related through fixed axis, similar to how Neva^{Uşşak} is linked to Neva^{Hicaz}. An important point to note is that there can be instances of double-transformations, such as the transition from Dügah^{Uşşak} to Neva^{Hicaz} , which are not directly related. Such transformations are also possible as can be seen within the progression of Bayâti Makam (Figure 2).

Based on this model we have compiled two extensive databases. The first one includes the most commonly used Perde^{Çeşni} structures in Turkish makam music and described them according to their perde content as well as the primary and secondary axes. The second one includes makam informations, in which makams are considered as Perde^{Çeşni} progressions.

3. THE MAKAMNETZ APP

3.1 Designing MakamNetz Virtually

Our data model for the pitch and intervallic content in music uses a textual approach to account for the variability of makam systems, with a focus on capturing trajectories and pathways between melodic units and makam structures. We leverage a relational database and the Django framework to effectively represent and analyze the intricate textual data structures, revealing the topological properties inherent in the makam universe.

The model's fundamental building block is "Perde," linking melodic units, while incorporating essential melodic components ("Cesnis") that establish connections within the makam network, providing insights into the evolution and complexity of the makam universe across different historical periods. The foundation of our data structures relies on Django "models," a prominent feature that empowers us to create intricate and well-defined data abstractions. This enables us to design and manage complex relationships between various entities in the system effectively.

For the frontend framework, we opted for React, a widely-used open-source framework designed for constructing single-page web applications. React's modular structure revolves around components, which are reusable building blocks. This feature facilitates the development of scalable and maintainable web applications. Additionally, React boasts efficient rendering capabilities, contributing to the creation of fast and responsive web experiences.

At the core of our application lies a graphical interface known as "react-force-graph." Leveraging the power of Three.js, this interface enables the creation of visually appealing 3D and 2D force-directed graphs. The combination of React and Three.js provides a robust platform for generating captivating visual representations of complex data structures. By harnessing the data generated through Django, we were able to craft an engaging user interface, providing users with a seamless experience in exploring and constructing the universe. The current version of the application framework exhibits versatility in handling diverse data models and generating distinct networks in accordance with our theoretical framework.

3.2 The App

Our app showcases the Çeşni Space (Figure 5), offering a visualization of different substructural units, and their interconnections. The small nodes represent individual Perde^{Çeşni} units, while the larger ones symbolize the fundamental *perde* upon which they are built. Nodes sharing the same color originate from the same perde. This visualization enables users to explore diverse pathways within

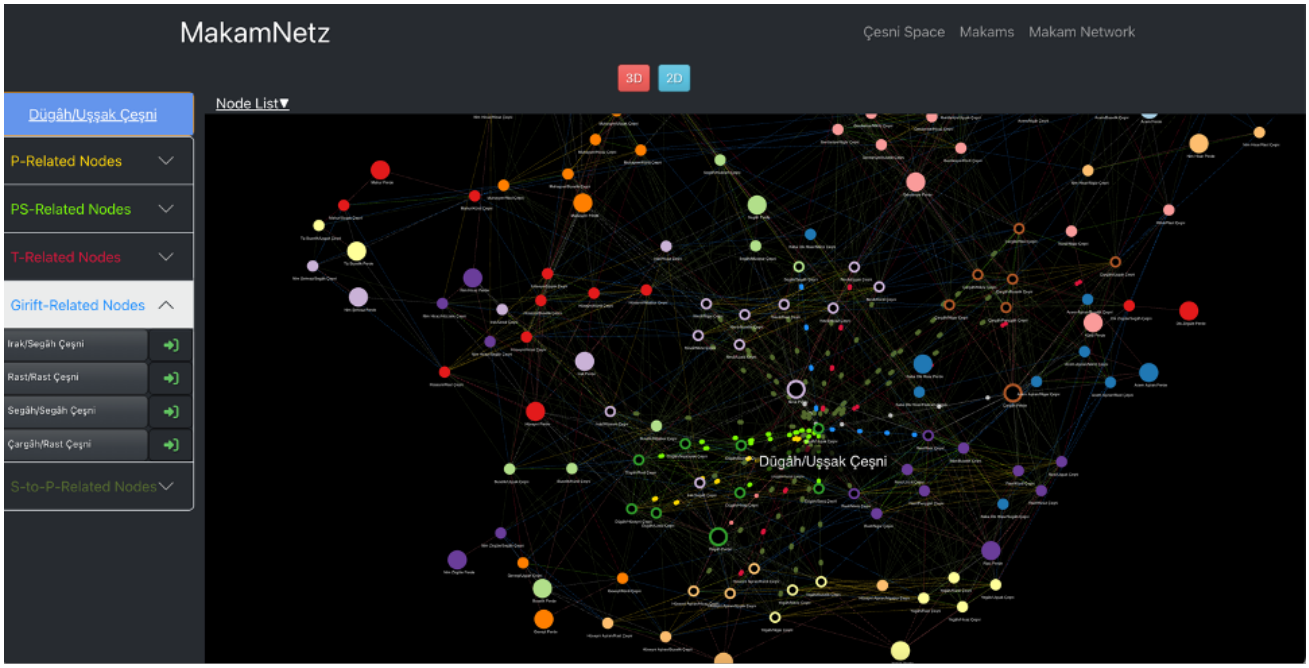


Figure 5. MakamNetz User Interface.

the network freely. Hovering over a node allows users to highlight all connected nodes and links, facilitating an in-depth analysis of transformations and relationships among different substructures. This feature aids in studying pathways in makams and gaining insights into the complex relationships between various structures.

In the 'Makams' section, users can delve into the melodic progressions within a single makam. The formal sections are color-coded, and the users can listen to how a selected makam sounds. Additionally, our app involves creating networks from multiple selected 'Makam Network. Presently, we are testing this feature with 17 makams. This subset allows us to observe and analyze relations between different makams and their sub-spaces. Our goal is to expand this functionality to encompass a larger number of makams that are currently in practice [4] .

3.3 ER Diagram and Relation Logic

MakamNetz primarily relies on the entity-relation (ER) model, which we transformed into a relational model using Django. Figure 6 displays the ER diagram, illustrating that the app is built upon four fundamental entities.

The smallest entity is Perde, where Makam pitch objects are encoded, each having a name and primary ID. The second entity, Perde/Cesni, consists of content that includes multiple Perde entities. We envisioned Perde/Cesni Content as a weak entity dependent upon Perde/Cesni. Each Perde/Cesni is related to another of its kind via a relation that relies on the content. Additionally, each Perde/Cesni is categorized within a Formal Section, which comprises three subcategories of the *seyir* (Initiation, Progression, Closure). This categorical hierarchy is encoded via an ISA relation. These formal units come together to compose a Makam, each of which has a unique ID and name. The diagram also illustrates the hierarchical relationship between each sub and superstructure through participation constraints encoded via thick lines. Each superentity has a participation constraint, as it cannot exist without the subentity. For instance, the Makam entity cannot exist without the formal section and thus relates to it via mandatory participation.

Much of relational logic is predicated on hierarchy; however, one particular configuration stands out: the self-relating Perde/Cesni entity, which constitutes one of the most crucial aspects of the application. This relationship, termed "Related to," relies on an algorithm to ascertain the nature of the relation, namely, p-axis, ps-axis, s-axis, and girift. The first three are contingent upon the common axis relation existing between two nodes under consideration, while the latter necessitates an object known as a girift chain to be formed. A girift chain embodies a predetermined scale-like structure composed of cyclically alternating cesnis. Within this framework, cesnis that can transition into one another via sliding one or two steps up or down are deemed girift-related. Given the finite number of girift chains, we have chosen to create an external JSON document rather than defining a separate entity. Formally, we can define the axis relations as follows:

Definition 1:

Let P_{MAKAM} be the set of all perdes and C_{MAKAM} be the set of all cesni sequences used in Turkish Makam.

- An axis perde in a cesni sequence is a perde that serves as a primary or secondary pivot point in the sequence.

[4] Demo videos of MakamNetz environment at work can be found in the MakamNetz YouTube page (https://www.youtube.com/@MakamNetz)

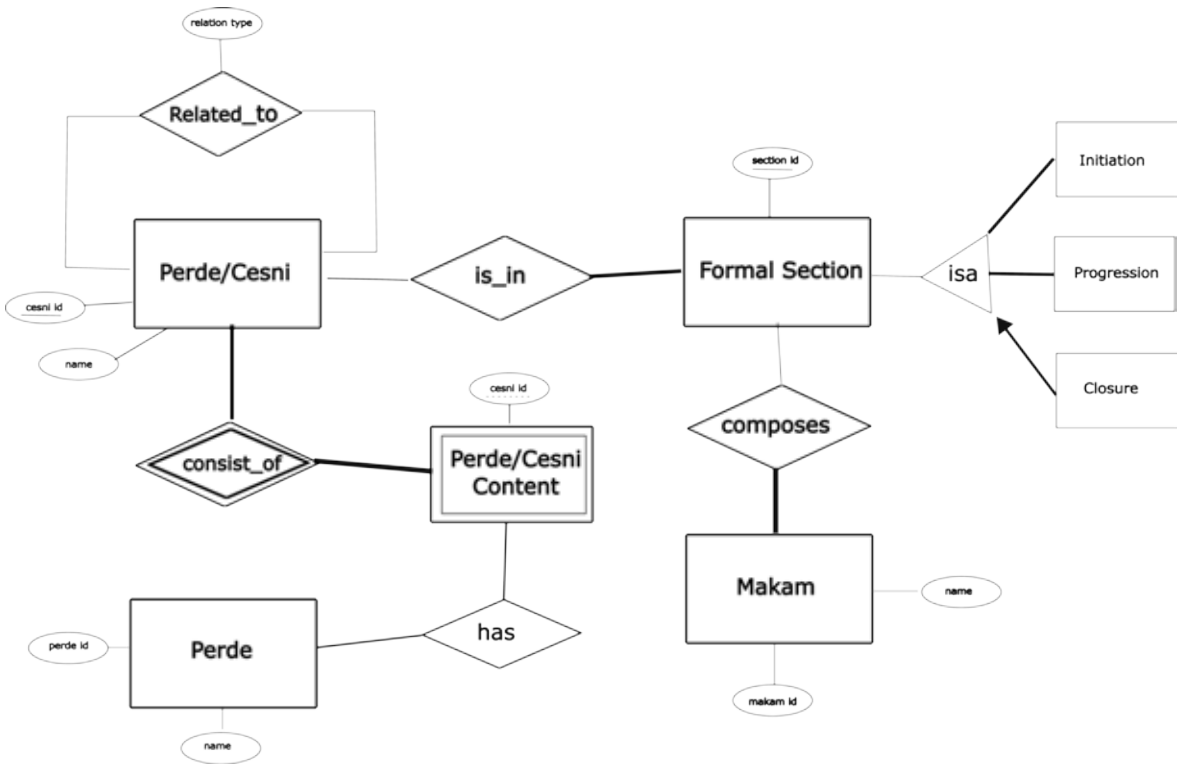


Figure 6. MakamNetz User Interface.

- A cesni sequence is a specific arrangement of perdes that define a makam's characteristic melodic progression.

$$A_P = \left\{ p \in P_{\text{MAKAM}} \;\middle|\; \begin{array}{c} p \text{ is an axis perde in some cesni} \\ \text{sequence in} \\ C_{\text{MAKAM}} \end{array} \right\}$$

Thus, $A_P \subseteq P_{MAKAM}$.

We define a matrix $PerdeMatrix$, where each element X_{ij} represents the relationship between two axis perdes p_i and p_j in terms of the cesni sequences they participate in. Specifically, X_{ij} is a set of cesni sequences that use both p_i and p_j as axis perdes.

Definition 2:

Define the set P_M as:

$$P_M = \{X_{ij} \mid p_i, p_j \in A_P \text{ and } X_{ij} \subseteq C_{MAKAM}\}$$

Within this framework, we can define:

- p-axis: The set of all cesni sequences where the primary axis perde is common.
- s-axis: The set of all cesni sequences where the secondary axis perde is common.
- ps-axis: The set of all cesni sequences where both primary and secondary axis perdes are common.

These can be defined as follows:

$$ps\text{-axis} = \{X_{ij} \mid X_{ij} \in P_M \text{ and } X_{ij} \neq \emptyset\}$$

$$p\text{-axis} = \bigcup_j X_{ij} \text{ for a fixed } i, \text{ where } X_{ij} \neq \emptyset$$

$$s\text{-axis} = \bigcup_i X_{ij} \text{ for a fixed } j, \text{ where } X_{ij} \neq \emptyset$$

4. DISCUSSION

While the model represents a significant advancement in the study of Turkish makam music, it is important to acknowledge certain limitations and considerations for future development. One notable limitation is the variability in the *perde* degrees within certain *çeşni* structures, such as Hicaz, which are influenced by their melodic relationships within specific makam contexts. Several studies have addressed this issue using computational methodologies, providing case-study frequency-band measurements of perde degrees derived from live recordings. However, there has been limited progress in general theoretical models concerning this matter, and there is no universal agreement on the most effective model. Since the discussion of this theory versus practice issue is beyond the scope of this research, the model framework does not accommodate such nuances arising from frequency-band differences of *perdeler* in different contexts.

Continued refinement of the transformational and relationship algorithms, alongside enhancements to visualization features, will be crucial in augmenting the usability and effectiveness of the application as a research tool. This includes a more accurate demonstration of the distances between substructures. Although the network model offers valuable insights into the harmonic connections and melodic pathways between Perde^{Çeşni} structures, distance calculation remains an unresolved issue. Currently, there is limited research on the similarity of Perde^{Çeşni} structures, which raises questions about their potential influence on

the smoothness of transformation. Thus, our model assigns equal weight to paths of equivalent total distance, irrespective of their practical applicability due to variations in frequencies. This is an area that the forthcoming phases of our research aim to address and improve in order to capture the nuances inherent in Turkish makam music.

5. CONCLUSIONS

In conclusion, MakamNetz provides a novel approach to mapping harmonic relations of Perde^{Çeşni} substructures and demonstrating transitional pathways within the makam universe, including individual spaces, neighboring constellations, and ostensibly remote systems. We believe that this work will offer a valuable tool for scholars, students, and enthusiasts of makam music alike, fostering exploration, discovery, and a deeper understanding of this rich musical tradition.

Acknowledgments

This work is supported by the Scientific and Technological Research Council of Türkiye, TÜBİTAK, Grant 122K923.

6. REFERENCES

[1] R. Cohn, "Introduction to neo-riemannian theory: A survey and a historical perspective," *Journal of Music Theory*, vol. 42, no. 2, pp. 167–180, 1998.

[2] ——, "Tonal Pitch Space and the (neo-)Riemannian Tonnetz," in *The Oxford Handbook of Neo-Riemannian Music Theories*. Oxford University Press, 12 2011.

[3] Kemânî Hızır Ağa, *Tefhîmü'l-Makâmât fî Tevlîdi'n-Nagamât*. İstanbul Topkapı Palace Museum Library.

[4] B. Bozkurt, "An automatic pitch analysis method for turkish maqam music," *Journal of New Music Research*, vol. 37, no. 1, pp. 1–13, 2008.

[5] Yahya Burak Tamer and Barış Bozkurt, "A microtonal tuning method: Test and discussion on the use of microtonal intervals for the performance of traditional turkish music," *Journal of Interdisciplinary Music Studies*, vol. 7, no. 1-2, pp. 73–88, 2013.

[6] B. Bozkurt, "Pitch Histogram based analysis of Makam Music in Turkey," Apr. 2014. [Online]. Available: https://doi.org/10.5281/zenodo.1211458

[7] Barış Bozkurt, M. Kemal Karaosmanoğlu, Bilge Karaçalı, and Erdem Ünal, "Usul and makam driven automatic melodic segmentation for turkish music," *Journal of New Music Research*, vol. 43, no. 4, pp. 375–389, 2014.

[8] B. Bozkurt, R. Ayangil, and A. Holzapfel, "Computational analysis of turkish makam music: Review of state-of-the-art and challenges," *Journal of New Music Research*, vol. 43, 03 2014.

[9] E. Benetos and A. Holzapfel, "Automatic transcription of Turkish microtonal music," *The Journal of the Acoustical Society of America*, vol. 138, no. 4, pp. 2118–2130, 10 2015.

[10] B. Bozkurt, "Computational Analysis of Overall Melodic Progression for Turkish Makam Music," Apr. 2014. [Online]. Available: https://doi.org/10.5281/zenodo.1211460

[11] S. Şentürk, "Computational analysis of audio recordings and music scores for the description and discovery of ottoman-turkish makam music," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2016.

[12] A. Karakurt, S. Şentürk, and X. Serra, "Morty: A toolbox for mode recognition and tonic identification," in *Proceedings of the 3rd International Workshop on Digital Libraries for Musicology*, ser. DLfM '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 9–16.

[13] B. Bozkurt, M. K. Karaosmanoglu, and C. Akkoc, "Weighing diverse theoretical models on turkish maqam music against pitch measurements a comparison of peaks automatically derived from frequency histograms with proposed scale tones," *Journal of New Music Research*, vol. 38, pp. 45–70, 4 2009.

[14] S. A. Shumays, "Maqam Analysis: A Primer," *Music Theory Spectrum*, vol. 35, no. 2, pp. 235–255, 10 2013.

[15] E. Bayraktarkatal and C. Güray, "Proposing a makâm model based on melodic nuclei," *Analytical Approaches to World Musics*, vol. 11, no. 1, pp. 1–45, 2023.

MODULE-LEVEL MIDI

Sean LUKE[1], **John TUFFEN**[2], and **Mathias BRÜSSEL**[3]

[1]*Department of Computer Science*, **George Mason University**, Washington, DC, USA
[2]**wonkystuff**, York, UK
[3]**tangible waves**, Murnau, Germany

ABSTRACT

We describe a set of conventions to run MIDI over the cables between individual modules in a modular synthesizer. Our approach allows modules to modulate one another; or to be collectively controlled by the musician; or both. The method deals with a variety of special problems arising from this fine-grained use of MIDI, such as module polyphony, MPE incorporation, parameter spaces, and avoiding full graphs and MIDI merge. We detail and discuss the issues and challenges, then offer a specification and provide usage examples.

1. INTRODUCTION

Module-level MIDI implements MIDI 1.0 [1] at the per-module level in a modular synthesizer. We envision MIDI as just another cable signal, integrated with gate / control voltage or audio. The goal is to create a flexible way for modules to control one another, to be collectively controlled by the musician, or both. Modules send MIDI to one another over standard modular cables rather than a bus, though a bus is not precluded.

Why would one wish to contaminate modular with MIDI? First, MIDI greatly simplifies the wiring for *polyphony* and *multitimbralism* in a modular synthesizer, and enables the use of *MIDI Polyphonic Expression*. Second, it helps automate the control of modules with *large numbers of parameters*. Third, it enables *patch storage and recall*. Fourth, it provides stable *pitch control* and a pitch standard with *microtuning*. Fifth, it provides *compatibility* with the external world.

In this paper we present three major challenges to implementing module-level MIDI, and how we overcome them. We then present conventions for control of modules by one another, or by an external controller. Our goal is to both provide foundations to be used by modules *now*, and a roadmap for more elaborate applications in the future. This roadmap can be modified as necessary later. Thus the spec is only version 0.5.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

Modular synthesizers, as originally proposed by Harold Bode [2], consist of multiple hardware *modules* that create or modify audio, or produce analog gate or *control voltage* (CV) to modulate other modules' settings. Audio or gate/CV flow from module to module via *patch cables*.

MIDI, or other forms of digital control of modules, used to be anathema in the modular world, but over the last few years it has seen significant growth in interest. The most common use of MIDI in a modular setup is modules that translate 5-pin DIN or TRS MIDI[1] into gate/CV for other modules. This configuration is meant to allow a musician, via an external controller, sequencer, or DAW, to send notes and parameter changes to select modules in real time.

Digital control of or among modules in modular synthesizers has to date largely used internal bus protocols. The forerunner in this area has been Buchla, whose systems feature an internal bus with multiple virtual lines meant to allow modules to send MIDI to one another or to receive MIDI from an external controller [3].

MIDI has also been run in a limited fashion on top of the internal gate/CV bus in Doepfer's Eurorack modular format. This scheme, developed by a group of Eurorack developers, is known as *Select Bus*, and has been primarily used to broadcast Program Change (PC) messages to modules to change their presets [4]. It is incompatible with other manufacturers' use of the gate/CV bus, including Doepfer's. Expert Sleepers has also introduced modules (such as the CVM-8 [5]) with one or more separate, dedicated internal Transistor-Transistor Logic (TTL) serial MIDI bus connectors.

Eurorack has also seen *I2C* used as an alternative bus transport. I2C is a two-wire packet-oriented communication scheme common in electronics. As popularized by the Monome Teletype [6], a simple control protocol runs on top of I2C from a leader module to some number of downstream follower modules. The protocol runs at 100kbps. Unfortunately it has not been standardized, resulting in a plethora of competing I2C-based messages among the modules which have adopted it. Some Buchla designs also employ I2C.

Most bus designs assume a single leader per bus. This means that a single bus line cannot easily support the scenario where module A controls modules B and C while B also controls C. To realize a general graph structure of control would require many real or virtual bus lines (as

[1] While TRS MIDI uses similar cables, it is not compatible with Eurorack as it requires specific circuitry, including an optoisolator.

in Buchla) and complex addressing. More elaborate bus protocols, essentially network protocols such as CANBus that define full graphs, have been occasionally proposed but not commonly adopted.

Beyond busses, other digital control schemes have focused on easing the use of polyphonic modules. Notably, TipTop has recently introduced *ART* [7], an event protocol running at 1.25 MBPS over standard cables in Eurorack. ART is accompanied by *Polytip*, a format for running parallel analog signals (importantly audio) using a cable with a plug identical to USB-C, but not using the USB-C protocol.

MIDI is also often used to interconnect and control individual guitar pedals on a pedalboard: this is somewhat analogous to modules in a synthesizer. Many guitar pedals accept MIDI, and certain pedals (an example would be those from Morningstar [8]) are meant to distribute PC or CC messages to them.

3. MIDI OVER MODULAR CABLING

In 2022 wonkystuff [9] introduced MIDI running as TTL serial over the 0.1" Dupont cables used to connect modules in the AE Modular synthesizer format [10] alongside gate/CV and audio. This scheme employs a channel distribution module to break TRS MIDI into separate per-channel message streams, each of which can then be routed to chains of MIDI-controlled oscillators, triggers, and gate/CV generators. A single chain of modules could be treated as a one monophonic synthesizer; and parallel chains could collectively implement polyphony via MPE [11] and other ways.

AE Modular makes this particularly easy because it is strictly 0–5V with a common ground, and standard MIDI is likewise also nominally 0–5V. AE Modular cables are single wires with ground on the power bus, and so are sufficient to run MIDI as TTL serial data.

wonkystuff's approach is primarily meant to allow the musician to control multiple modules using a single external controller, sequencer, or DAW. However it has certain limitations. First, it does not have any way to distinguish among the modules listening in on the same channel. This is particularly problematic if one wanted, for example, three oscillators on a single channel but with different parameter settings, such as detune. Second, it does not allow modules to *modulate one another*: for example, a MIDI-controlled envelope module modulating a MIDI-controlled oscillator module, while both are controlled by a DAW. Third, it does not address the issues raised by MPE, nor MIDI-controlled polyphonic modules, a popular recent trend.

In this paper we significantly extend this approach, proposing a set of MIDI conventions to follow in order to address these and other issues, while staying compatible with MIDI standards. We have expended effort in making basic forms of the protocol easy for simple devices to implement, while providing more capabilities to more sophisticated modules and systems.

3.0.1 Why Not Just a Bus?

We are interested in running MIDI over cables, rather than a bus, because it makes it much easier to do complex topologies, and because it does not require memorization of virtual "behind the scenes" routings. It does, of course, require more visible wires. We must admit that because of these very wires, MIDI over cables is more tangible and retains much of modular's physical charm: it's just another cable routing like CV or audio.

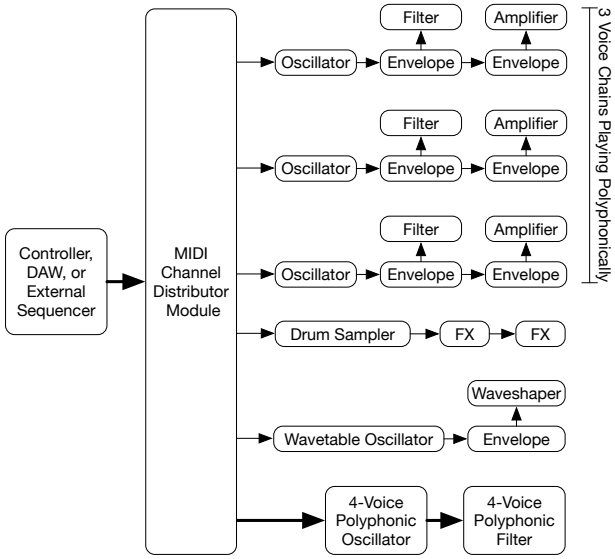


Figure 1. Over-Elaborate MIDI Connectivity Tree Example. The Distributor module breaks out MIDI data by channel to individual voice chains. Thick arrows indicate single MIDI connections containing multiple channel voice data. Note the three identical polyphonic voice chains, the separate wavetable voice chain, and the *two* FX modules in the Drum Sampler chain. Finally, note that the envelopes are organized as non-leaf nodes, to enable optional MIDI Injection.

That being said, a bus is not in any way precluded: indeed we have discussed the possibility of combining cabling with a MIDI bus internally on AE Modular.

3.0.2 Why MIDI 1.0?

MIDI 1.0 is now 40 years old, and its failings are very well understood. But its advantages are also compelling. MIDI has extensive tooling available, is flexible and extensible, and is a simple and easily parsed serial protocol. But most importantly, it is omnipresent in the music industry. By adopting conventions layered over MIDI, we immediately gain access to a huge number of devices and software.

Why not MIDI 2.0? MIDI 2.0 addresses MIDI 1.0's primary flaws, but it is a large and complex protocol that may be difficult, if not impossible, for small processors to implement. It is not yet widely adopted in the industry, and is immature. It would seem to be a poor option for modular systems with simple modules.

Last, proprietary protocols running on top of I2C have seen increasing popularity in certain Eurorack circles; but they are neither standard nor standardized, and often are little more than arbitrary MIDI translations.

4. CHALLENGES

We begin with three challenges in digital control of modules via MIDI, and how we will address them. The first challenge is **topology**. MIDI is meant for one controller to control multiple downstream devices. We want to allow modules to control one another, but we want to avoid a full graph structure and MIDI Merge. The second challenge regards **namespaces** and **voices**. MIDI has a single namespace: channels. But when issuing parameterized control, we need to distinguish between receiving modules based both on the voice they are assigned to and the particular module in that voice. We also consider polyphony and polyphonic modules. The third challenge lies in making certain that the changes we make in the first two challenges do *not* require a monolithic, complex protocol that cannot be implemented by **simple devices**: we need a way to ramp up in complexity as needed.

4.1 Challenge 1: Topology and Speed

Anyone who has connected many hardware synthesizers is painfully familiar with the complexity of wiring them. The large number of modules in a modular synthesizer can make this even uglier. Ideally we would place all the modules on a network and allow any module to communicate with any other, though this would be complex to implement, require memorization, and lose the immediacy and (in truth) the charm of modular.

Regardless, MIDI is far from a network: it is a unidirectional, daisy-chained communications protocol. We abandon a bus and allow MIDI to assume a tree topology, with a leader at the root sending messages to all nodes in the tree. The root is likely a controller, DAW, or hardware sequencer, under the control of the musician. Non-leaf nodes in the tree both respond to and forward these messages to their children. Figure 1 illustrates an (overly elaborate) example of a tree of MIDI control among modules.

In a full graph, allowing any node to talk with any other node would require MIDI Merge. We permit MIDI Merge but actively avoid requiring it due to its complexity and introduction of cycles. But we can claw back some of this power by allowing modules to modulate nodes in their subtrees. We do this by optionally introducing a "lightweight merge" in the form of *MIDI injection*.

Injection is simple: before it passes data from its parent to its children, a module is permitted to *inject* some MIDI messages into the stream to send to one or more of its children. This would allow Module A to modulate Module B, and B to modulate C, while all of them are being controlled by the root. Generally non-leaf nodes would often be modulators (envelopes, say), and modulation targets would be leaf nodes. Figure 2 illustrates MIDI injection.

MIDI is also slow: it runs at 31250 bits per second. [2] But assuming MIDI does not exit the modular synthesizer, there is no reason why it couldn't run as fast as we would like internally. 115200 BPS seems reasonable as it is a common baud rate. Speed would be constrained by the maximum speed afforded by the medium (patch cables) and the maximum rate that the basic modules' microcontrollers can handle.

[2] You may be interested as to why. Early MIDI synthesizers had CPUs operating at multiples of 1 MHz. To quote: "The 31.25 kHz clock can also be obtained from hardware, for example, by dividing 1 MHz by 32." ([12], page 7, §"Hardware").

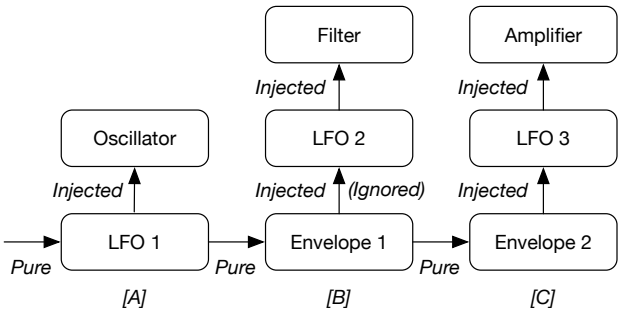


Figure 2. Three Injection Examples. [A] LFO 1 injects modulation data into the stream to control the Oscillator pitch, while passing "pure" non-injected data to Envelope 1. [2] Both Envelope 1 and LFO 2 modulate the Filter. This is done by Envelope 1 injecting modulation data to LFO 2, which ignores it and passes it on while also injecting its own data. Envelope 1 passes "pure" data to Envelope 2. [C] Envelope 2 injects data to modulate LFO 3's depth. LFO 3 responds to this modulation data, and removes it from the stream before injecting its own data to modulate the Amplifier.

MIDI traditionally incurs latency as data is transferred from module to module. Part of this is due to its use of optoisolators (which we do not need). But excess MIDI injection could also have significant latency costs.

Last, bidirectionality is useful for transferring and storing presets. While a MIDI tree is unidirectional, there's nothing stopping modules from providing a single back-channel wire to provide bidirectionality, at least between two modules.

4.2 Challenge 2: Voices and Namespaces

If we are to control parameters of multiple modules, we will have a namespace problem.

Historically MIDI has had just one limited namespace to distinguish between devices: its 16 MIDI channels. However the recently adopted MIDI Polyphonic Expression (MPE) specification also uses this namespace to distinguish among polyphonic voices in the same device, and has introduced "MPE Zones" in a halfhearted attempt to allow the two usages to coexist [11].

Inside a modular synthesizer we are faced with a bigger namespace issue. First, as polyphony and multitimbrality are now popular in modular synthesizers, we must distinguish among voices. Second, we must distinguish between different modules in a given voice: an oscillator will have different parameters than a filter.

To unravel this, we suggest adopting the approach suggested by MPE: commonly each channel is a separate voice, [3] and the subtree of modules which collectively pro-

[3] There are exceptions. For example, a drum module would likely produce different drum sounds per-note on the same channel.

duce that voice are known as a *voice chain*. Within a voice chain, modules may be distinguished from one another by their *ID*, a value ≥ 1. As all modules in a given voice chain will be listening in on the same channel, IDs are then used to divide up the parameter space in CC and NRPN. Figure 1 illustrates the need for this namespace, as it has polyphonic voice chains, multiple envelopes per chain, and multiple effects units.

A module might set its ID via a DIP switch, or via a value stored in EEPROM, and very trivial modules might be simply fixed to an ID, though this is discouraged. NRPN is straightforwardly partitioned according to ID. But CC is very limited, and so we have attempted to divide it optionally and flexibly by ID. Modules are encouraged to use certain regions in CC according to their ID but we allow much leeway. If necessary, and if NRPN is not feasible, a module may allocate additional IDs for more CC space. We have also provided a "poor-man's NRPN", called *auxiliary parameters*, for IDs without CC allocations (Section 5.5).

In some cases, modules need both separate parameters and joint parameters. For example, two oscillators might wish to share the Pulse Width parameter but have different Detune parameters. A module is permitted to share portions of its ID parameter space with another.

4.2.1 Polyphony and Polyphonic Modules

Several models of polyphony are achievable. First, there is standard MIDI 1.0 *one-channel polyphony,* where a *voice assigner* module would distribute note messages to individual voices, as well as copying voice-parameter CCs to all voices. This could also be encapsulated as a single *polyphonic module*. Second, a *polychain* is a string of monophonic oscillators, each of which passes note data along to a subsequent oscillator via soft MIDI THRU when it is already playing a note. Polyphony in this case would likely be limited by latency. Third, in *MPE* each note is pre-distributed to its own channel by a controller. The distributor would then send each channel to a separate voice chain, or would group several channels (hence voices) to send to a polyphonic module.

A polyphonic module may be thought of as several monophonic modules glued together, each in its own voice chain. A polyphonic module could sport one MIDI input per voice, or a single MIDI input for multiple notes along a single channel, or one input for multi-channel data. In the second case, it might terminate multiple monophonic voice chains (as a multi-VCA package, for example). See Figure 1 for an example of polyphonic modules in a MIDI tree configuration.

4.2.2 Why not Distinguish Modules by Channel?

There are many reasons not to assign each module in a voice chain to its own channel, not the least of which is incompatibility with MPE. But one issue is that notes are sent to specific channels. In a modular synthesizer several modules may need to respond to the same note, such as oscillators responding to pitch, envelopes or LFOs responding to gate, etc. But they cannot do this if they are on different channels. Channels need to be associated with voices.

4.3 Challenge 3: Simplicity versus Versatility

We imagine that for most people, needs will be very simple, and many modules will have small microcontrollers. Thus we need a protocol that can be both simple or more sophisticated and versatile depending on need and cost. We address this in five primary ways.

First, we assume that MIDI voice data for a given voice chain has been filtered to a single channel by a distribution module, and thus most modules in a chain may operate as OMNI. Polyphonic modules may instead benefit from MIDI sent with multiple channels.

Second, modules have default ID settings and these defaults vary by module type (oscillator, filter, etc.) Thus modules often can be simply plugged in and played in basic configurations. Trivial modules need not have a user-settable ID at all (though this is discouraged) if there are a small number of them and their default IDs are all different.

Third, simple devices need only respond to basic 7-bit CC parameters, but 14-bit CC and NRPN are optionally provided for more elaborate parameters. CC parameter regions are set up to be flexible and not set in stone.

Fourth, many systems will just be controlled externally, and not need module-to-module MIDI modulation: and so basic modules could ignore this facility. But module-to-module MIDI modulation is made to be easy: modulators need only be set to modulate one of 8 generic parameters that we have set aside in the CC space. Modulation targets would just map these to their first 8 CC, Auxiliary, or NRPN parameters.

Fifth, the use of MIDI does not preclude traditional CV in any way: they can be mixed without incident.

Along these lines, we can think of the protocol at different levels of capability, depending on need:

4.3.1 Level 1: Trivial Support and CC Learn

At its simplest, the protocol would support a voice chain for a monophonic synthesizer with a small number of modules and little external control by the user. An upstream MIDI distribution module would have filtered out undesired channels, so a downstream module can use OMNI. Such modules would have hard-coded IDs and only use the CC parameters allotted to those IDs. If two modules had conflicting hard-coded IDs, one could still have its CCs remapped to elsewhere in the CC space via a CC Learn function or a sysex message. Having hard-coded IDs is discouraged however.

A module would provide only "pure" (hardware) MIDI THRU. Polyphonic MPE-style voice chains would be handled entirely by the MIDI distribution module with multiple per-channel MIDI OUTs.

4.3.2 Level 3: Settable NRPN and Program Change

More advanced modules would have settable IDs, particularly if they wished to use the additional parameter space afforded by Auxiliary parameters or NRPN. These modules might also support presets via Program Change, and saving presets via Program Save (Section 5.5). Note that Program Change would affect *all modules* on a channel/voice chain that respond to it. An optional proposed feature, *Module*

Select (Section 5.4), would allow one to to indicate which modules should respond to a Program Change/Save message.

4.3.3 Level 3: Modulation Modules

Some modulators (envelopes, LFOs, sequencers, etc.) could change *other* modules' parameters via MIDI. These modules could simply change one of eight generic *modulation CCs*, and their modulation targets would map these CCs to 8 parameters of their choice. Modulators would perform *MIDI injection* to, and removal of modulation messages from, the MIDI stream.

4.3.4 Level 4: Polyphonic Modules and Merge

Polyphonic modules would support a single multi-channel MIDI connection to connect to other polyphonic modules, or optionally support multiple OMNI MIDI connections for simple voice chains. We might also see a MIDI Merge module if useful.

5. PROPOSED CONVENTIONS (V. 0.5)

This is v. 0.5, and is subject to further revisions.

The topology of the system is a tree. The root node is often a musician's controller, DAW, or external sequencer. Most remaining nodes are individual modules. Parent nodes in the tree are normally directly connected to child nodes via user-pluggable cables, though they could be connected via a bus. Data flows from parent to child only. We adopt the MIDI THRU concept: nodes can pass data from their parent on to their children.

Nodes normally pass data through a hardware MIDI THRU. Nodes wishing to modulate other nodes are also permitted to *inject* data into a *software* THRU, that is, add data to the stream passed to their children. Nodes may have multiple MIDI THRUs and they can be of different types with different kinds of injection: however they should always have at least one "pure" (non-injected) MIDI THRU. Nodes may remove data passed to them by their parent prior to injection of new data: see Section 5.3. See also Figure 2 for examples of injection and MIDI THRU.

A node may have multiple parents only if it keeps their incoming data streams separate (such as a polyphonic module), or if it fully implements MIDI Merge, a rare but permitted case. MIDI Merge plus MIDI THRU make cycles possible: cycles can overwhelm the MIDI stream and their impact on the topology is not defined. Nodes may to have additional MIDI connections outside the primary topology (such as for bidirectional patch transfer), but their effect is not defined here.

The speed of MIDI from module to module should be at least 31250 BPS, but can be faster than this if it is standardized as such on a given modular platform.

Nothing here forbids modules from responding to MIDI as they like, such taking control over one or more entire channels if they see fit. For example, it is entirely reasonable that a sophisticated polyphonic oscillator module might simply claim an entire channel and all note and CC events on that channel. However it is often the case that multiple modules must listen to the same channel and respond to notes and CCs in order to to work together to form a voice. They must do this without conflicting with one another. Other modules must control one another via MIDI while also being externally controlled themselves. The conventions below deal with cases such as these.

ID	*Type*
1	Oscillators, Samplers, Note→Gate/Trigger Modules
2	Envelopes
3	LFOs, Sequencers, Clock→Gate/Trigger Modules
4	Filters
5	VCAs, Mixers
6	Effects, Audio Processors
7	CC→CV Generators
8	Miscellaneous

Table 1. Default ID Types

5.1 MIDI Channels, Voices, and Voice Chains

Usually a MIDI channel is associated with a *voice*, that is, the set of modules which together produce a sound in response to a MIDI note. Some voices can have the same parameters and so collectively act polyphonically, while multitimbral voices may be distinct.

Commonly a distributor module receives MIDI from an external sequencer and breaks it out to MIDI outputs per-channel (thus per-voice). Each MIDI output is routed to a subtree of modules working together to produce one voice: this is a *voice chain.* As the channels have been broken out, modules in a voice chain may respond to MIDI voice messages in OMNI mode.

A *polyphonic module* may subsume multiple voices, hence multiple channels. Polyphonic modules may receive separate individual streams (as OMNI) and so participate in voice chains; or they may receive and send multiple channels via a single MIDI connection.

A distributor module receiving MPE ought to copy messages from the MPE master to the key channels as it deems appropriate.

5.2 Module IDs and Parameters

Each module in a voice (channel) has a unique ID 1–15, commonly 1–8. The ID determines the region of parameters allocated to the module in the CC, Auxiliary Parameters (Section 5.5), and NRPN spaces. Modules have default assignments based on the module type, shown in Table 1.

Modules are encouraged to have user-changeable IDs, but simple modules can have hard-coded IDs with default assignments under the (poor) assumption that the user only has one module of each type per voice. Indeed, if a module does not use CC, Auxiliary Parameters, nor NRPN, its ID is at present not relevant.

Module IDs 1–8 have parameters allocated as CCs. Each ID has two 14-bit CC pairs called "a" and "b", and five 7-bit CCs ("c" through "g") allocated. We have set some default ID types regions to include some classic CC names which show up in DAWs and controllers.

CC	Name	CC	Name	CC	Name	CC	Name	CC	Name	CC	Name
0/32	Bank Select	16/48	3a (a/i)	64	Sustain Pedal	80	3g	96	Increment	112	1g
1/33	Mod Wheel	17/49	3b (b/h)	65	1c Glide Switch	81	4c	97	Decrement	113	Mod. c
2/34	Open Breath Controller	18/50	4a (a/i)	66	1d	82	4d	98	NRPN LSB	114	Mod. d
3/35	Auxiliary	19/51	4b (b/h)	67	1e	83	4e	99	NRPN MSB	115	Mod. e
4/36	Open Foot Controller	20/52	5a (a/i)	68	Legato Switch	84	4f	100	RPN LSB	116	Mod. f
5/37	Glide MSB/Open LSB	21/53	5b (b/h)	69	1f	85	4g	101	RPN MSB	117	Mod. g
6/38	Data Entry	22/54	7a (a/i)	70	2c	86	5c	102	7c	118	Mod. h
7/39	Volume MSB/Open LSB	23/55	7b (b/h)	71	2d	87	5d	103	7d	119	Open
8/40	1a (a/i)	24/56	8a (a/i)	72	2e Release	88	5e Hi Res Vel	104	7e	120	All Sound Off
9/41	1b (b/h)	25/57	8b (b/h)	73	2f Attack	89	5f	105	7f	121	Reset Ctrlrs.
10/42	Pan MSB/Open LSB	26/58	Mod. a	74	MPE Timbre	90	5g	106	7g	122	(Reserved)
11/43	Expression Controller	27/59	Mod. b	75	2g Decay	91	6c FX1 Amt	107	8c	123	All Notes Off
12/44	6a (a/i) FX Ctrl 1	28/60	Open	76	3c Vib. Rate	92	6d FX2 Amt	108	8d	124	(Reserved)
13/45	6b (b/h) FX Ctrl 2	29/61	Open	77	3d Vib. Depth	93	6e FX3 Amt	109	8e	125	(Reserved)
14/46	2a (a/i)	30/62	Open	78	3e Vib. Delay	94	6f FX4 Amt	110	8f	126	(Reserved)
15/47	2b (b/h)	31/63	Open	79	3f	95	6g FX5 Amt	111	8g	127	(Reserved)

Table 2. CC Parameters. The first two columns are 14-bit CC pairs. Each module ID has two 14-bit CCs (a, b) and five 7-bit CCs (c...g). A 14-bit CC may be split into two 7-bit CCs: the parameter *a* may be split into *a* (MSB) and *i* (LSB), and *b* into *b* and *h,* hence the notation (a/i) and (b/h). Also shown are System, Open and (Reserved), and Auxiliary parameters, and Modulation CCs (a...h). ID default regions are set up to attempt to align with some CCs that have optional, traditional names. Open parameters may be used with care, with the understanding that other modules may also respond to them and that their purpose may change in the future. (Reserved) parameters may not be used.

A module is free to split one, or both, of its 14-bit CCs into 7-bit CCs if it needs 8 or 9 CCs. If just one is split, it should be "b", split into "b" (the former MSB) and "h" (the former LSB). If two are split, then "a" may also be split into "a" and "i" respectively. Table 2 shows the allocated regions for each ID.

Parameter space is allocated to IDs 9...15 as discussed in Section 5.5. Also, NRPN allocates 256 parameters to each ID, as shown in Table 3. The first 9 correspond to parameters "a" through "i" in the CC map allocated to the module. NRPN parameters are 14-bit even if their corresponding CC parameters are 7-bit. Reserved NRPN parameters should be ignored.

Modules may also change IDs and this ability is strongly encouraged. Some modules might plausibly use CC Learn (listening for an incoming CC and memorizing it) or other remapping scheme to relocate their CCs to regions normally allotted to other (missing) modules. However in this case they still ought to have user-changeable IDs if possible, or at least a fixed ID for default CC values.

A module may be assigned more than one ID if the need arises. Additionally, in some cases two or more modules may wish to share parameters in common in addition to having separate regions. For example, two oscillators may wish to both be changed by pulsewidth modulation, but have detune changed separately. This should be done by having one module set to respond to certain parameters of the other, ideally by ID.

A polyphonic module could have different IDs for each voice, but normally they would be the same ID.

A module might also perform all the functions of a voice and not need other modules. Such modules are effectively synthesizers in and of themselves and may be assigned a full channel per voice: they may disregard the suggested parameter assignments described here, and do as they wish with CCs and with NRPN, though they might wish to retain them for consistency.

NRPN Region	*Range*
0–255	[Reserved]
256–511	The first 256 parameters for ID 1
...	
3841–4097	The first 256 parameters for ID 15
4098–16383	[Reserved]

Table 3. Allocated NRPN Regions

5.3 Modulation CCs

Two 14-bit CC parameters and six 7-bit CC parameters are designated *Modulation a...h*. These generic parameters may be injected by modulation modules such as envelopes, LFOs, or sequencers and sent to their children. A child responds to these CCs as if they were its first eight CC or NRPN parameters (a..h). If a module does not implement that many parameters, it may ignore the remaining modulation parameters.

Modules are free to ignore this feature. Otherwise they are encouraged to have, as a switchable option, the ability to respond to or to ignore incoming modulation CCs. Modules that modulate other modules via MIDI injection *must* have this option. If a modulation module itself responds to a particular modulation CC, then it must remove that CC from its stream before injecting new messages. See Figure 2 for examples of injecting data and and ignoring the same.

The two 14-bit modulation CCs are 14-bit even if the corresponding regular CCs are 7-bit. They have special rules to improve atomicity in 14-bit CC. When an MSB arrives, the parameter is set (to MSB$\times$128) only if the prior CC was also an MSB for the same parameter, or if there has been no prior CC. An MSB must precede any series of one or more LSB. When an LSB arrives, the parameter is then changed to MSB$\times$128+LSB. Modules may inject parameters other than modulation CCs, but should not remove those.

MSB	*Function*
0	Program Save
1	Current Program Save (= 0), Revert (= 1) [Reserved (LSB > 1)]
2–15	[Reserved]
16–31	The first 16 parameters for ID 9
...	
112–127	The first 16 parameters for ID 15

Table 4. Auxiliary Parameters. LSB is the value.

5.4 Program Change and Bank Select

Modules may respond to Program Change messages alone; or include Bank Select (always providing MSB and LSB). Modules responding to Bank Select should do so within some contiguous range $b \in 0...M$ where $b = 128 \times MSB + LSB$. If a Bank Select value is out of range for a given module which supports it, it should also ignore the Program Change. Program and Bank Select may be injected and removed.

Optional Module Select We are considering using the MSB (0) of Bank Select as *Module Select*, used to state *which module IDs* should respond to Program Change, Save, and Revert messages, as shown in Table 5.

5.5 Auxiliary Parameters

The 14-bit CC pair 3/35 provides certain *auxiliary parameters*, as described below. CC 3 (the MSB) sets the parameter number as shown in Table 4, and CC 35 then sets its value. A CC 3 must appear before any series of CC 35: a bare initial CC 35 should be ignored.

CC 3 = 0 is a *Program Save* message. CC 35 (LSB) then gives the value of the program (0...127). The default is 0. This tells a module to save its current state to the specified program in the current bank.

CC 3 = 1 is a directive to *Save to* (value = 0) or *Revert to* (value = 1) the *current* program.

The CC 3 parameters ≥ 16 indicate the first 16 parameters (a...p) for IDs 9...15, and are set only as 7-bit. These parameters should correspond to the first 16 parameters for these IDs in NRPN. CC 35 (LSB) gives their value. To set them in 14-bit, use NRPN.

Reserved parameters and values should be ignored.

We recognize that *Program Save* and *Current Program Save* would represent a break from MIDI tradition in that, excepting System Exclusive directives, there presently exist no MIDI messages which modify the semi-permanent state of the synthesizer, that is, memory that survives power cycling. Thus these directives may be considered still under consideration.

5.6 System and Reserved Parameters

There are sixteen 14-bit and 7-bit CC parameters reserved for system purposes as shown in Figure 2. These include Bank Select, Modulation Wheel, Data Entry, Sustain Pedal, All Sound Off, and so on. All modules may respond to these parameters as appropriate.

The 23 reserved CC parameters should be ignored.

Value	*Function*
0	All Modules on Channel (the default)
1–15	IDs 1 through 15 respectively
16–127	[Reserved]

Table 5. Optional Module Select Values

5.7 Clock, RPN, Sysex, and MIDI Modes

Modules may remove and inject clock signals to perform clock division or multiplication. RPN may be responded to by all modules as appropriate. We note that RPN MPE Configuration parameter may be of particular use to a top-level distribution module. Modules may also respond to System Exclusive messages.

Some modules or voices play notes simultaneously (such as a drumkits), and others play monophonically. The behavior in question depends on the choice of modules for a voice. For this reason, MIDI Modes do not make sense in this context. Modules may be thought of as being in Modes 3 or 4: but at any rate, they should not respond to Mode change CC messages, which are reserved.

6. EXAMPLES

6.1 Example: Trivial Monophonic Oscillator

Consider a square wave oscillator with pulse width controlled by CC. It has sound out and gate out. It cannot be tuned and does not respond to pitch bend. This module would have one MIDI IN and one pure MIDI THRU, with no injection. The module would use OMNI as its channel and could be hard-coded to ID 1, but settable ID would be better. It could respond to Note On/Off and have a CC set to PW/Wave. A better oscillator would also respond to NRPN for PW/Wave.

6.2 Example: Injecting Monophonic LFO

Consider a simple LFO that responds to rate, depth, and type, and that can inject MIDI to a target. It responds to aftertouch to increase the depth, in addition to a CC. It also responds to the Mod Wheel to add to the rate. The module would have one MIDI IN, one pure MIDI THRU, and one MIDI THRU with injection.

Via a switch, we would stipulate which modulation parameter was output, and whether the module ignored (passed) or removed modulation data sent to it by its parent. Via a 3-DIP switch we can set the module ID. It would use OMNI for its channel.

This module would respond to note on, aftertouch, mod wheel, and several mappable CCs for rate, depth, and type depending on the ID, as well as NRPN parameters for the same. It would also respond to (and remove) Modulation CCs for rate, depth, and type.

6.3 Example: Polyphonic Wavetable Oscillator

Finally consider a 4-voice wavetable oscillator, with four independent oscillators on four different voices. The oscillator can respond to pitch, velocity, bend, and portamento. We can set the detune and degree of aliasing in the

wavetable. The oscillator has patches and so responds to Program Change but not banks.

The module would have one MIDI IN and THRU that apply to all channels as well as, at its discretion, four MIDI IN/THRU socket pairs to work in conjunction with monophonic voice chains. Each oscillator could have its own ID, but normally they would use the same ID. The module would respond to note on and off, mappable CCs for Osc Tuning, PW/Wave, Wavetable, and aliasing, as well as All Sound Off, All Notes Off, and NRPN. It would also respond to Program Change.

7. DISCUSSION AND FUTURE WORK

This proposal is version 0.5. Beyond the MIDI modules already available commercially, we have implemented preliminary versions of this proposal in hardware, including CC/NRPN response and message injection or removal. We expect to need to make changes to the proposal, and publish several revisions.

We recognize that one issue with this proposal is that, in order to deal with the dual-namespace issue, it must use much of the CC parameter space. There are only 128 CCs to go around. We have tried to be flexible and to allow NRPN and Auxiliary Parameters, but may need to add additional tricks in the future. The proposal also does not define a full graph structure. We have attempted to allow modules to modulate one another while remaining true to MIDI 1.0 standards, and permit (but avoid) MIDI Merge; but we recognize that there may be rare complex configurations which are awkward to implement in our scheme.

This proposal was developed for and by the AE Modular community but it is not in any way limited to AE Modular. Nothing in the spec precludes porting to other platforms, like Eurorack. We very much would like to see it used on those platforms, or inspiring similar designs.

Acknowledgments

Our thanks to Beppe Sordi, Luca Ludovico, and the AE Modular community for their help, criticism, and feedback, and to LIM at U. Milan for its support.

8. REFERENCES

[1] *The Complete MIDI 1.0 Detailed Specification* (Version 96.1, Third Edition), MIDI Manufacturers Association, Los Angeles, 1996.

[2] H. Bode, "A new tool for the exploration of unknown electronic music instrument performances," *Journal of the Audio Engineering Society*, vol. 9, no. 4, pp. 264–266, 1961.

[3] Buchla Electronic Musical Instruments, "User's guide for the 200e Electric Music Box (v1.4)," buchla.com/guides/200e_Users_Guide_v1.4.pdf, 2017, as of 1/1/2024.

[4] "Eurorack modular system preset protocol," docs.google.com/document/d/1YhPvAI6oliwLYSHhDholdAUym-TfBDGlxST-zDCP-qw, 2016, as of 1/1/2024.

[5] "CVM-8 firmware v1.1 user manual," www.expert-sleepers.co.uk/downloads/manuals/cvm-8_user_manual_1.1.pdf, 2022, as of 1/1/2024.

[6] "Monome Teletype," monome.org/docs/teletype/, as of 1/1/2024.

[7] "ART: Chords, melodies, harmony, and polyphony in Eurorack," tiptopaudio.com/art/, as of 1/1/2024.

[8] "Morningstar Engineering," www.morningstar.io, as of 1/1/2024.

[9] J. Tuffen, "MIDI: a new signal type for AE," wonkystuff.net/midi-a-new-signal-type-for-ae/, 2023, as of 1/1/2024.

[10] "Tangible Waves (AE Modular)," www.tangiblewaves.com/, as of 1/1/2024.

[11] *MIDI Polyphonic Expression (Version 1.0)*, MIDI Manufacturers Association, Los Angeles, 2018.

[12] S. Jungleib, *The Complete SCI MIDI*, Sequential Circuits, Inc., 1983.

MIDI 1.5, So to Speak

Sean Luke[1] and **Luca Ludovico**[2]

[1]*Department of Computer Science*, **George Mason University**, Washington, DC, USA
[2]*Department of Computer Science*, **Università degli Studi di Milano**, Milano, Italy

ABSTRACT

In this position paper we argue that there may yet be more to squeeze out of MIDI 1.0 before adopting MIDI 2.0 wholesale. As MIDI 2.0 has been slow to roll out, it is worthwhile considering what kinds of backward-compatible improvements could be made to MIDI 1.0 as a stopgap to address some of the issues which drove MIDI 2.0 in the first place. In this paper we provide examples of six major improvements that could be made, as a proof of concept and strawman argument.

1. INTRODUCTION

In 1981 Ikutaro Kakehashi contacted Tom Oberheim, suggesting an international communication standard for music synthesizers, and Oberheim then recommended that Dave Smith be included in the informal discussion group. Smith's resulting proposal, called the Universal Synthesizer Interface, or USI, was first discussed at the Gakki Fair in 1981, and then after feedback from Japanese manufacturers it was presented at the following AES Conference [1]. After a few more iterations, Smith renamed the protocol MIDI and presented it at the NAMM show in 1982. Initial resistance from manufacturers was eventually won over, and MIDI became the de facto standard for control of electronic musical instruments [2].

That was over 40 years ago. MIDI has been very successful, and its weaknesses are also now well understood. In 2020 the MIDI Manufacturer's Association (MMA) formally adopted MIDI 2.0, a collection of six documents totaling 349 pages representing a major collaborative effort to update MIDI 1.0 for the modern age [3]. MIDI 2.0 tries to address most of MIDI 1.0's major problems and to provide room to grow.

MIDI 2.0 has been slow to roll out. Operating systems have only recently introduced MIDI 2.0 drivers despite likely having had significant runway prior to the 2020 announcement. Our private discussion with some major hardware manufacturers has not revealed enthusiasm for the protocol. Relatively few products have to date adopted MIDI 2.0, and almost no hardware devices have done so. This has been blamed on industry slowness due to the COVID pause, or the time needed to build drivers or software, and this may in fact be so. We have been assured by a member of the MIDI 2.0 committee that 2024 will be 2.0's *Annus Mirabilis*, and in truth we very much hope that it is!

However, it also may not be. We suspect that the chief cause of MIDI 2.0's slow adoption to date is simply that it is large and complex. And so we have explored what might be done in the interim. If MIDI 2.0 requires further revisions, what might be used as a stopgap measure? If small hardware manufacturers or users reject it for its complexity, what might be provided instead, to smooth the ramp until adoption?

Thus this position paper poses the question: **what would a MIDI 1.5 look like?** Are there improvements that we can *still* add to MIDI 1.0 which would be backward compatible and yet provide some of the features long sought after by MIDI 1.0 critics?

Though what follows might look like a specification proposal, it is in fact simply a strawman exercise, meant to show that many improvements *could* be made to MIDI 1.0 without jumping to the full features, and complexity, of MIDI 2.0, and to suggest that it might be worthwhile considering a stopgap measure if MIDI 2.0 needs time to live up to its promise.

1.1 Previous Work

1.1.1 Previous extensions and alternatives

While MIDI 1.0 never changed its version number prior to 2.0, it has had a great many revisions, and presently stands at Version 4.2 [4]. There have been many proposed extensions to, transports for, and alternatives to MIDI 1.0 for notation, song organization, performance, and music analysis: we refer the reader to *Beyond MIDI* [5] for a good summary of many proposals up to 1997 alone. Probably the most important alternative to MIDI 1.0 as a communications protocol has been *Open Sound Control* (or *OSC*) [6], which defines a full graph topology much like MIDI 2.0, albeit with an emphasis on portable and human-readable addressing and communication.

1.1.2 Previous Use of Reserved MIDI Status Bytes

Some of the solutions we propose use reserved MIDI status bytes. There were no doubt many previous proposals to use the four reserved status bytes, though perhaps not in response to MIDI 2.0! Here are two, both concerning the System Real-time message `F9`.

In 2016 Michael Lauter published an online whitepaper entitled "The case for MIDI real-time MARK message F9h" [7]. In it he argued that `F9` could be used to mark the

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

start of a new measure or musical phrase to make it easier to sync disparate devices.

The second is more curious. *F9* has been incorrectly used as a *MIDI Tick* message to be emitted every 10 milliseconds to allow for a simple form of absolute time sync. This nonstandard usage has spread to various applications due to its inclusion in the standard Arduino MIDI library [8]. It is not clear what the origin of this usage was, but the earliest reference we can find appears to be from a MIDI fan website [9].

We have been informed that the MIDI 2.0 Committee also considered using status bytes early on, among many options, but rejected them. Among the chief reasons for rejection was the discovery that there existed hardware interfaces in the field, or operating system drivers or libraries, that did not behave properly when confronted with unexpected use of the reserved status bytes. Ultimately the committee chose to develop an entirely new protocol.

We find this argument unpersuasive. It is essentially saying that, because *some* nonconforming interfaces and software would need to be revised to work properly with Protocol A, we should abandon it and move to Protocol B, where *all* of them would need to be not only revised but redesigned from scratch. There are many compelling reasons to move to a better protocol: but we think this is not one of them.

We do note that a few specifications external to MIDI proper, such as MIDI RTP [10], have over the years improperly *banned* reserved status bytes rather than reserved them, and so might require revision.

2. MIDI 1.0 DESIDERATA

MIDI's failings are well known. Here are a few.

MIDI is Slow MIDI is a serial spec that runs at 31250 BPS, including a start and stop bit. This was chosen carefully: it is exactly 1/32 of 1MHz. In fact, originally it was to run at 19200 BPS until revised after suggestions from Japanese manufacturers [2]. It was thought that 31250 would be fast enough to "prevent objectionable delays between equipment" [11, p.7].

Parameter and Value Resolution are Poor MIDI pitch, velocity, and aftertouch resolution are only 7-bit: though the MIDI Tuning Standard allows for pitch changes with $100/2^{14}$ cents resolution. The value and parameter resolution of MIDI CC are both only 7-bit, and much of the CC parameter space is used up by official conventions and standards. One alternative route, 14-bit CC, increases, both the value space but dramatically reduces the parameter space. Another solution, RPN/NRPN, increases the value and parameter spaces to 14 bit, but is slower.

MIDI has Namespace and Polyphony Problems MIDI has a single namespace, namely its 16 channels, which is very limited from a modern perspective. This is remediated to some degree in software by increasing the number of MIDI ports or devices, each with its own set of 16 channels, but this is a poor fix. To make matters much worse, MIDI channels are not fully polyphonic, notably in per-note parameters and pitch bend. MPE was proposed to fix this, but its approach (usually making each channel a monophonic voice) means that MPE synthesizers can respond properly to at most 15 notes per MPE Zone, and there can be at most two of them, dividing those 15 notes amongst themselves. Furthermore in doing so this radically eats into the channel space allotted to other synthesizers.

Limited Discovery and Capability Inquiry MIDI only has a primitive discovery mechanism. Via system exclusive, devices can inform us that they exist but not what protocols they can respond to.

Patch Transfer is Undefined MIDI never defined how to transfer patches or update fined-grained patch data outside of CC or NRPN. Manufacturers, left to their own devices, have constructed a zoo of incompatible, buggy, and often profoundly ill-conceived patch transfer or revision schemes [12].

MIDI is Unidirectional MIDI was designed for a single device to send messages to multiple downstream devices. It was not meant for devices to communicate with one another in an arbitrary graph topology.

2.0.1 A Few Solutions

We are not so grandiose as to suggest that solutions to all these problems can be found with small modifications to MIDI 1.0. Certainly speed, for example, cannot be improved in general as MIDI 1.0 on serial hardware is fixed to 31250 BPS. However a number of improvements can be made while retaining backward compatibility with properly conforming devices. We refer to these improvements colloquially as *MIDI 1.5*.

MIDI 2.0 is also touted as backward compatible with MIDI 1.0: but only in the very weak sense that if the older device appears to not understand it, one could always revert to 1.0 instead. It cannot intermesh with properly conforming 1.0 devices. We think it is better placed in the "clean break" category along with approaches such as Open Sound Control [6].

How might we make MIDI 1.5 backward compatible with 1.0? First off, 1.0 has two options for official extension, Registered Parameter Numbers (RPN) and Universal System Exclusive messages. Less well known is that MIDI 1.0 also has *four* status bytes still not in use. Two of these status bytes (*F9* and *FD*) are System Real-time messages and so are generally expected to be only one byte in length. The other two (*F4* and *F5*) are System Common messages and do not have defined lengths. These status bytes could be defined for our purposes.

In this paper we outline six ways in which we could improve some of these issues while staying backward-compatible with MIDI 1.0, by using Universal System Exclusive and RPN messages and certain reserved status bytes. They are:

Discovery Given a bidirectional channel, we can determine if downstream devices are MIDI 1.5-Ready.

Sectors These are roughly analogous to MIDI 2.0's groups. This scheme extends MIDI's channels to 2048.

MPE Improvements We employ Sectors to extend MPE instruments to have more than 15 notes and more than two MPE zones.

Ports This scheme can be used instead of, or in addition to, CC or NRPN. Compared to NRPN, Ports have more parameters and often faster speed. Ports also fix NRPN's lack of atomicity, a major failing.

Patch Transfer We can use MIDI 2.0 Capability Inquiry (CI) to query devices for capabilities and parameters, and update the same, as is uses System Exclusive. But MIDI 2.0 CI is heavyweight, and so we also propose a basic patch dump standard.

Pitch Resolution We could improve the per-note pitch resolution with a CC.

3. DISCOVERY

We begin with methods to determine if a downstream device exists and is *MIDI 1.5 Ready.*

A few preliminaries. First, it has been said that MIDI 1.0 is unidirectional. But that is not really true! Bidirectionality just requires two MIDI cables. The fundamental distinction is that in MIDI 1.0, a device may not be the recipient of messages from two different senders (absent MIDI Merge). However MIDI 1.0 is perfectly capable of device discovery and capability inquiry if two cables or virtual devices are used. We presume bidirectionality for MIDI 1.5 Discovery, and later for Capability Inquiry and the Patch Dump standard (Section 7). However this is not absolutely necessary: if we only have a unidirectional channel, then we can still proceed with many MIDI 1.5 capabilities with care.

Second, several MIDI 1.5 messages are System Exclusive Common messages. These messages have a specific format: *F0 Type DeviceID SubID ... F7. Type* is 7E for non-real-time and 7F for real-time message categories. The *DeviceID* is set on the Device itself: all downstream devices have different DeviceIDs. A DeviceID of 7F means "all devices". The *SubID* is the namespace for each category of System Exclusive Common messages. MIDI 1.5's namespace is not yet defined: we will use the placeholder *M1.5* throughout.

3.0.1 MIDI 1.5 Readiness and Tolerance

A *MIDI 1.5 Ready* device understands MIDI 1.5 and how to respond to MIDI 1.5 properly. Using MIDI 1.5 Discovery, we can probe to determine if devices are MIDI 1.5 ready. In order to take advantage of MIDI 1.5, the endpoint devices must be MIDI 1.5 Ready.

A *MIDI 1.5 Tolerant* device is a device which is not MIDI 1.5 Ready, but ignores MIDI 1.5 messages and passes them through unmodified. In order to use MIDI 1.5 data, all devices must be at least MIDI 1.5 Tolerant. In particular, all drivers, interfaces, routers, and other in-between devices must be MIDI 1.5 Tolerant. MIDI 1.5 uses reserved status bytes from MIDI 1.0. It does not use any System Real-time status bytes. Thus it should be transferred to downstream devices by properly-adhering MIDI 1.0 operating system drivers, MIDI interfaces and routers, and so on. However there exist ones which filter out these status bytes, convert them, or otherwise behave badly when receiving them, and so will interfere with 1.5 even if the downstream device is MIDI 1.5 Ready. Thus we may query a downstream device and have it respond that it is MIDI 1.5 Ready, only to find that we cannot send 1.5 to it because of non-MIDI 1.5 Tolerant operating system driver.

It's possible for multiple devices to be on the same MIDI daisy-chain, and one is MIDI 1.0 (but 1.5 Tolerant) while another is MIDI 1.5 Ready. This would require that the MIDI 1.5 Ready device restrict itself to only operate within the channels available to it.

3.1 Existence Query and Response

We use MIDI System Exclusive (or *Sysex*) Device Inquiry to determine if bidirectional communications is working, and which devices are capable of responding. There may be more devices listening than are capable of responding. To send a Sysex Device Inquiry to a specific device, we send *F0 7E DeviceID 06 01 F7*.

Ideally (and always if the Device is MIDI 1.5 Ready) the Device should respond with a Sysex Device Inquiry Response: *F0 7E DeviceID 06 02 Manufacturer (1–3 Bytes) Device (4 Bytes) SoftwareRevision (4 Bytes) F7*. The *Device* consists of a 2-byte family code and a 2-byte family member code. We will use the *DeviceID, Manufacturer,* and *Device* later.

3.2 MIDI 1.5 Declaration and Query

The sender can declare that it is itself MIDI 1.5 Ready by broadcasting *F0 7E 7F M1.5 00* to all devices. Note that the sender does not itself have a DeviceID, though we could perhaps adapt 7E as a convention.

We next query a given device as to whether it is MIDI 1.5 Ready or not, with *F0 7E DeviceID M1.5 01 F7*. If the device is MIDI 1.5 Ready, it should respond with *F0 7E DeviceID M1.5 00 F7* .

If a device has responded with the Sysex Device Inquiry Response and a MIDI 1.5 Declaration, then we assume the following:

- The device exists.
- The device can respond to us.
- The device is MIDI 1.5 Ready: this includes being able to respond usefully to MPE messages.

If the device has responded with a Sysex Device Inquiry Response but not a MIDI 1.5 Declaration, then we should assume that the device is *not* MIDI 1.5 Ready, though it may be MIDI 1.5 tolerant and able to respond usefully to MPE Messages.

If the device has not responded with a Sysex Device Inquiry Response, then it may not exist, or may or may not be MIDI 1.5 Ready but is unable to respond to us. We must tread carefully.

4. SECTORS

Sectors are roughly analogous to MIDI 2.0's groups. They effectively extend the number of channels to 2048 and ultimately allow MPE to have more than 15 notes.

There are many reasons to want more than 16 channels. But one critical one is MPE, which has co-opted MIDI channels to provide per-voice parameterization. But because there are only 16 channels, MPE can support at most 15 voices. Furthermore, because MPE uses MIDI channels in the way it has, they can no longer be used in a multitimbral fashion if MPE is applied to all 15 voices.

4.1 Approach

A Sector is a group of 16 channels. There are 128 Sectors in all. Sectors can be used to provide 2048 individual streams of multitimbralism, and to support up to 256 MPE devices. Non-voiced messages are applied to all Sectors. All voiced messages are interpreted as applying only to the current Sector.

The current Sector is set with the Sector Declaration, *F5 Sector*. The default Sector is Sector 0, the "MIDI 1.0 Sector". Because there may exist MIDI 1.0 devices listening, Sectors must be careful to interoperate with them. To do this, the Sender must know (or suspect) which channels are being used by MIDI 1.0 devices, and assume that these channels are *always* Sector 0 regardless of the Sector declaration.

4.2 Sector and Channel Discovery

A MIDI 1.5 Ready device should respond to a Sector and Channel Request *F0 7E DeviceID M1.5 03 F7* to state which Sectors and channels it responds to. The response is *F0 7E DeviceID M1.5 02 [Sector Channel MPEInstrumentTag]* F7*. The *MPEInstrumentTag* is the MPE Instrument Tag of the given Channel: see Section 5. If *MPEInstrumentTag* is 7F, then the Channel has no MPE Instrument assigned, and is a plain Channel.

4.3 Alternatives

Sector Declarations could be less efficiently done in System Exclusive as *F0 7F 7F M1.5 00 SectorByte*.

4.4 Issues

Whereas declaring the channel of a voiced message is 3 bits, changing the Sector requires two bytes. Ideally per-Sector messages would be grouped together when possible to require only one Sector change message.

If a MIDI 1.5 Ready receiver has not heard a previous Sector message (perhaps he started listening late, or noise corrupted the message), he will apply follow-on messages to the wrong Sector. This is analogous to similar synchronicity issues in RPN and NRPN.

4.5 Mapping to MIDI 2.0

MIDI 2.0 introduces 16 *groups*, each of which may hold 16 channels, for a total of 256 channels. These could map to the first 16 Sectors. MIDI 2.0 CI also is compatible with groups, and thus we believe with the first 16 Sectors.

5. MPE IMPROVEMENTS

MPE is a hack that uses channels to enable per-note parameterization. Unfortunately, in doing so, MPE has three major limitations. First, it is limited to only fifteen voices. Second, it is limited to only two MPE instruments (that is, *MPE Zones*). Third, use of MPE consumes channels which could otherwise be usefully used by other devices.

We addressed the third issue with Sectors, discussed in Section 4. We address the other two here.

Our approach extends MPE Zones to *MPE Instruments*. An Instrument is a collection of MPE Zones grouped together into one voice pool. An MPE Instrument may have up to 128 voices shared across a number of MPE Zones. Each MPE Instrument has an *MPE Instrument Tag* and MPE Zones are each assigned a tag to indicate that they are part of a given Instrument. There can be up to 127 Instruments.

Because they use so many channels, realistically if there are any MPE devices listening, they must either all be, or none of them must be, MIDI 1.5 Ready. We further would recommend that MIDI 1.5 Ready devices only use MPE Low Zones.

5.1 MPE Capability Discovery

A device should respond to a MPE Capability Request to indicate which instruments it has available. The MPE Capability Request is *F0 7E DeviceID M1.5 05 F7*. The device would respond with *F0 7E DeviceID M1.5 04 NumMPEInstruments [MPEInstrumentTag MaxVoices]* F7*. Here, *NumMPEInstruments* indicates the maximum number of MPE instruments that may be created for the device. $MaxVoices+1$ is the maximum number of voices per instrument. This, combined with the Sector and Channel Request (Section 4.2) gives us which instruments are meant for which channels.

With or without discovery as useful additional information, we can proceed to define MPE Zones, per Sector, using the standard MPE RPN messages.

5.2 Issues

In MIDI 1.0, downstream devices decide on their own channels, but in MPE it is the *sender* which decides the structure and size of the MPE zones available, and the devices are left to fend for themselves after being told this structure. This may become more problematic with increasing numbers of channels, and zones.

6. PORTS

Ports are an attempt to achieve better speed than NRPN in certain common situations, to provide better parameter and value resolution than NRPN, and to overcome mistakes in the RPN/NRPN specification.

What mistakes? Since its inception [11] CC has been conventionally divided into two regions: 7-bit CC messages offering a resolution of 128 values, and optional CC message pairs offering 14-bits, or a range of 16384 values. In

the second case, a "most significant byte" (MSB) message is paired with a "least significant byte" (LSB) message. The number formed would be MSB×128+LSB. For example, to send a 14-bit value for parameter 2, you'd send the MSB of the value as CC 2, and the LSB of the value as CC 34. The MIDI designers felt that it was reasonable for a simple controller to just send the MSB, and so argued that the LSB should be thought of as an optional "fine tuning" of the MSB value. This was accompanied with the following text:

> If 128 steps of resolution is sufficient the second byte (LSB) of the data value can be omitted. If both the MSB and LSB are sent initially, a subsequent fine adjustment only requires sending the LSB. The MSB does not have to be retransmitted. If a subsequent major adjustment is necessary the MSB must be transmitted again. When an MSB is received, the receiver should set its concept of the LSB to zero.

This was a mistake. First off, a parameter's value range may not be 0–16383, but might more often be, say, 0–194. Following these instructions, manufacturers would have to awkwardly "stretch" the 194 range to 16384 so as to allow a bare MSB to be sent with no LSB. Many chose not to, and so the MSB would only be (in this example) 0 or 1, and the LSB would be the dominant factor.

Second and more problematic, imagine if the value is currently MSB=4, LSB=9. A new MSB=7 arrives. Is an LSB soon arriving to accompany it? If the controller were just a 7-bit controller, maybe not. The device is required to assume, for the interval between an MSB and an LSB, that the value is MSB×128+0, which is *wrong* if an LSB is arriving. 14-bit CC is not atomic.

These were reasonable oversights in the first MIDI draft, but then they were needlessly inherited in NRPN and RPN. Some manufacturers (like Sequential!) have abandoned NRPN "fine tuning" as unworkable, but others have retained it. Controller and DAW vendors must now offer both "fine" (LSB-primary) and "coarse" (MSB-primary) versions of 14-bit CC and NRPN.

Atomicity could have been fixed in 14-bit CC and RPN/NRPN by requiring that after an MSB the value not be changed until the next LSB appears. But that ship has sailed. We could have changed it here too, but it would not be backward-compatible with MIDI 1.0.

6.1 Approach

We introduce 65536 *Ports*, parameters like RPN/NRPN. A Port can be set to 16384 values.

Ports provide the full parameter or value in a single message, and so they are atomic. There is no notion of "coarse" versus "fine tuning", as neither the MSB nor LSB can be provided by itself. Unlike RPN and NRPN, ports can also auto-increment the parameter in order to easily perform dumps of parameter ranges.

Port parameters and values are set with the same message: *F4 Command MSB LSB*. If this message is meant to change the *Port parameter*, then *Command* takes the form *00hhcccc*, where *hh* are the high bits of the parameter, added before the MSB and LSB for a total of 16 bits or 65536 parameters; and *cccc* is the channel.

If the message changes the *Port value*, then *Command* takes the form *01ppcccc*. The MSB and LSB define the value with 14 bits, thus 16384 values. The *cccc* is the channel. The *pp* state how the value is to be set:

- 00 Set the value to the given amount.
- 01 Set the value to the given amount, then increment the current parameter. 65535 increments to 0 (wrapping around).
- 10 Increment the value by the given amount, with a ceiling at 16383.
- 11 Decrement the value by the given amount, with a floor at 0.

6.1.1 *Efficiency Comparisons*

The worst-case scenario in overhead for setting values in a Port require also changing the Port parameter each and every time. This would be a total of 8 bytes. But once a Port parameter is declared for a given channel, values may be repeatedly changed in that channel. Normally this would be 4 bytes per change. However if we were to extend running status to include F4, changing values in a Port would be 3 bytes amortized.

In comparison, the worst case scenario for NRPN is [Param-MSB Param-LSB Value-MSB Value-LSB]*. This is 12 bytes. The amortized case for NRPN is [Param-MSB Param-LSB Value-MSB Value-LSB [Value-MSB/LSB]*]*. Assuming running status, and that the user only changes the LSB (or only the MSB) this requires 2 bytes per change (else 4 bytes).

Ports have an advantage when making *bulk* changes to many parameters in a row. Ports can auto-increment the parameter, and so bulk changes are 3 bytes per change; whereas for NRPN they are 12 bytes.

Thus Ports are 1.5× faster for single parameter changes and 4× faster for bulk changes of multiple parameters, but are roughly 1.5× slower for multiple changes to the same parameter, such as a filter sweep. [1] There is of course no reason why NRPN and Ports could not be interchanged, assuming that their parameter numbers mapped to the same parameters.

7. PATCH TRANSFER

One of the MIDI's biggest failings is its lack of a standardized patch dump or editing standard. This is curious as the system exclusive message was, from the very beginning,

[1] We considered using *F4* to change the Port parameter as *F4 MSB LSB*; the value would be sent using Polyphonic Aftertouch. Port parameter 0 (the default) would be standard Polyphonic Aftertouch. This would give 16383 parameters, would yield a 2× improvement for single parameters and a 6× improvement for bulk changes. To do this trick we'd need *some* three-byte voiced channel message, and Polyphonic Aftertouch was the clear candidate as changing it would have the least impact and MPE forbids its standard use. However it could still cause problems with non-MIDI-1.5-Ready devices.

primarily meant for patch dumps [11]. As patch dumps were never standardized, this has led to a menagerie of patch dump protocols which can only be described as incredible in their variance and poor design [12]. Yet at the same time, MIDI proposed a standard *sample dump* protocol despite MIDI being a poor choice for sample dumps due to its speed.

It may be too late to propose a patch dump protocol, for MIDI 1.5 or even 2.0, because a number of manufacturers have begun to treat system exclusive as a trade secret or to abandon it entirely in favor of alternative USB-based byte protocols [12]. This is in stark contrast from the past, where manufacturers proudly published their system exclusive specifications in the name of openness that was the hallmark of MIDI.

That being said, a standard protocol would go a long ways towards fostering a better ecosystem. To this end, we propose the following set of conventions:

7.0.1 Command Syntax

System exclusive commands take the form *F0 Manufacturer (1–3 bytes) Payload...*, followed by a status byte, usually *F7*. *Manufacturer* is 1–3 bytes and sets the manufacturer ID, and thus defines a namespace. Manufacturer IDs are registered with the MMA.

Our proposed commands take one of two forms:

- Short Commands
  Header... Command (1 byte) Data (0–4 bytes) F7
- Long Commands *are usually of the form*
  Header... Command (1 byte) Length (2 bytes) Encoding Data... (any length) Checksum F7

...where the *Header...* is *F0 Manufacturer (1–3 bytes) Device (4 bytes) Sector*. The Device is four bytes, comprised of the device's family code and family member code, 2 bytes each. The *Sector* is the Sector (or just *a* Sector) to which the device is listening, or is 7F, that represents any Sector to which the device is listening.

The *Command* is the directive in question: the first 32 commands are reserved and 10 are defined as described below. The rest are up to the manufacturer's discretion. The *Data* is encoded using the given *Encoding* as described in Section 7.1. The *Length* is the length of the Data only, as a 14-bit number, and the *Checksum* is the sum of all the bytes, unsigned, of all bytes in the Payload except for *F0* and the Checksum, mod 128.

Patch data which uses the conventions in this Section should be fully, freely, and openly specified.

7.0.2 Actions

A Device must be able to issue or respond to the following twelve actions. Individual parameters and patches *could* be dumped with Ports, but we recommend the Dump commands here for a consistent sysex interface.

Version and Configuration This would be *Header... 00 Encoding Data... F7*. The specification for the *Data...* in this message must be publicly available.

Request Version and Configuration This would be *Header... 01 F7*. The device would respond with a Version and Configuration Message as described earlier.

Send Current Patch Load the patch into current working memory such that it can be used immediately by the musician. Do not write to patch memory, nor perform a program change. This would be *Header... 02 Chunk Of Length Encoding Data Checksum F7*. The data may be broken up into multiple pieces. *Of* indicates the number of pieces in total, and *Chunk* $(0...Of-1)$ represents the which piece this message is. Chunks must be sent in ascending order and do not take effect until the final Chunk is received.

Request Current Patch Request the patch presently in current working memory, that is, the one currently being used by the musician. Do not perform a program change. This would be *Header... 03 F7*. The device would respond with a *Send Current Patch* message.

Dump Patch Write the following patch to a given location in patch memory. Do not perform a program change to the patch. This would be *Header... 04 Chunk Length Encoding Location Data Checksum F7*. Note that the data may be broken up into multiple pieces. *Of* indicates the number of pieces in total, and *Chunk* $(0...Of-1)$ represents the which piece this message is. Chunks must be sent in ascending order and do not take effect until the final Chunk is received.

Request Dump Patch Request a patch from a given location in patch memory. Do not perform a program change. This would be *Header... 05 Location F7*. The device would respond with a *Dump Patch* message.

Change Patch This consists of an optional *Bank Select* message, followed by *Program Change* to any bank and patch. A Bank Select is sent with CC 32 for the LSB and CC 0 for the MSB; both the MSB and LSB must be sent, and the bank number is interpreted as MSB $\times 128+$ LSB. If there is only one bank, then Bank Select may be omitted.

Request All Patches in Bank This would be *Header... 06 Location F7*. Do not perform a program change to any patch. The device would respond by dumping each patch in turn from the current bank.

Change Mode If the Device is modal, the mode is changed. For example, a synthesizer might be playing in single-mode, multitimbral-mode, or drum mode, each with its own set of patches. After changing mode, the device may default to whatever patch in the mode it sees fit. This would be *Header... 07 Mode F7*.

Stop All Sound This kills all sound from the device, and is done with CC 120. The device is encouraged to also implement CC 123 but it is not required to.

Dump One Patch Parameter This is *Header... 09 Parameter (4 bytes) Value (4 bytes) Checksum F7*, where both the *Parameter* and *Value* are 4 MIDI bytes (the *Value* is encoded via Encoding #2 in Section 7.1); or by *Header... 09 Length*

Parameter (4 bytes) Encoding Value... Checksum F7, where the *Value* is encoded the provided *Encoding* as discussed in Section 7.1. Manufacturers are also encouraged to set parameters, to the degree that they can, via Ports or NRPN.

Query One Patch Parameter We query a parameter via *Header... 08 Parameter (4 bytes) Checksum F7*, where the *Parameter* is 4 MIDI bytes. The device responds with a Dump One Patch Parameter message.

7.1 Encodings

We propose five standard data encoding methods. The first three are obvious to anyone who has dealt with patch dump system exclusive messages in the past. The fourth is a logical extension of them. The fifth encoding is meant for situations where the size of some elements, or of every element, may vary arbitrarily.

0. Each element, 7 bits long, encoded in one byte.
1. Each element, 8 bits long, encoded by sending groups of 7 bytes at a time without their high bits, then a final byte with all 7 high bits. In this final byte, the low bit is the high bit of byte 1.
2. Each element, 14 bits long, encoded in two bytes, LSB sent first and MSB sent second.
3. Each element, 16 bits long, encoded by breaking it into two bytes, LSB first, then sending the resulting stream of bytes using Method 2.
4. Each element encoded in turn using Variable Length Encoding, as described below.

Prior to encoding, signed values should be converted to 0...*N* where 0 represents the lowest negative number and *N* represents the highest positive number.

7.1.1 Variable Length Encoding

There are many ways to do a variable length encoding. Here is one simple possibility that is easy and fast to implement. The encoding has a variable-length *header* followed by a certain amount of *data*. The header is zero or more 1-bits, followed by a single 0-bit, which together control the length of the data. The data appears immediately after this. The length of the message is equal to the number of bytes necessary to contain the header (padded with data bits), plus *N* additional bytes of data, where *N* is the number of 1-bits in the header. The encoding has at least an 85.7% efficiency.

Some examples. We pack up to 6 bits as 0dddddd, encoding the data *dddddd*. We may pack up to 12 bits as 10dddddd dddddddd, 18 bits as 110dddd dddddddd dddddddd, 24 bits as 1110ddd dddddddd dddddddd dddddddd, and so on. The header may cross byte boundaries, so we pack up to 42 bits as 1111110 [dddddddd]×6, but then 55 bits as 1111111 0dddddd [dddddddd]×7 and 61 bits as 1111111 10ddddd [dddddddd]×8.

The low bit of the data should start at the far end of the encoding and proceed towards the front, zero-padding at the front if necessary. For example, the value 371, encoded in binary as 101110011, should be converted to 12 bits as 000101110011, and then packed as 1000010 1110011. It is not necessary to provide the full encoding if a value can be packed in less. For example, if we have a 32-bit parameter but we are setting its value to just 23, we could just encode it as 0010111.

7.2 MIDI 2.0 Capability Inquiry

MIDI 2.0 introduced MIDI 2.0 Capability Inquiry (CI), which allows a sender to determine the capabilities of a downstream device, and to query and change its parameters, given bidirectional communication. Unlike much of the rest of MIDI 2.0, this protocol can be done entirely with MIDI 1.0 System Exclusive messages. For that reason we could also adopt it here, and it would serve to replicate some of the functionality discussed in this Section. We note however that it has a complex specification totaling 183 pages spread over four documents (M2-100-U through M2-103-UM of [3]).

8. PITCH RESOLUTION

MIDI's pitch resolution is per-note. If you want a pitch whose value is in-between notes, you have three options in MIDI 1.0. First you can use the *MIDI Tuning Standard* (MTS), which specifies which pitch each key falls on: but you are stuck with 128 pitches. Second, you can send pitch bend prior to playing the note: but this will change the pitch of all other notes. Third, you can do the same thing with MPE, as it requires that bend not affect released notes, and there is normally one note per channel: but now bend can no longer be used to affect released notes! And, of course, it requires that you use MPE.

Ideally, the high resolution pitch of a note would accompany the note itself. MIDI Manufacturer's Association document CA-031 [13] proposed that prior to playing a note, one might send CC 88 as its *high resolution velocity*, that is, as the LSB of the note velocity. We suggest selecting another CC to do the counterpart: *high resolution pitch*. The LSB could be measured in 1/128 of a semitone; or perhaps it would be easier to simply measure it in cents, reserving the top 28 values.

We note three downsides. First, this is two to three times slower than a plain Note On, depending on Running Status. Second, the receiving device would have to support this CC. Third, the resulting 2^{14} total pitch range is less than MIDI 2.0's 2^{32}, 2^{21} for MTS, or even 2^{15} for MPE's default pitch bend.

9. CONCLUSION

We hope that MIDI 2.0 will be successful, but it has had a difficult roll-out. It is a large specification imposed on a small community. We have argued that in the meanwhile, or in the event that 2.0 is not fully adopted, MIDI 1.0 could still be usefully improved while maintaining backward compatibility and relative simplicity. We admit that not everything can be improved, such as raw serial hardware speed. But enhancements forming MIDI 1.5, so to speak, could have a major impact well worth the small cost.

10. REFERENCES

[1] D. Smith and C. Wood, "The 'USI', or Universal Synthesizer Interface," in *Audio Engineering Society Convention 70*, Oct 1981.

[2] MIDI Manufacturers Association, "MIDI history chapter 6 — MIDI begins 1981–1983," https://www.midi.org/midi-articles/midi-history-chapter-6-midi-begins-1981-1983, as of 1/1/2024.

[3] *MIDI 2.0 Specification*, MIDI Manufacturers Association, 2023.

[4] *The Complete MIDI 1.0 Detailed Specification (Version 96.1, Third Edition)*, MIDI Manufacturers Association, 1996.

[5] E. Selfridge-Field, Ed., *Beyond MIDI: The Handbook of Musical Codes*. MIT Press, 1997.

[6] M. Wright and A. Freed, "Open SoundControl: a new protocol for communicating with sound synthesizers," in *International Computer Music Conference (ICMC)*, 1997.

[7] M. Lauter, "The case for MIDI real-time MARK message F9h," http://lauterzeit.com/tech/midi/case_for_MIDI_F9h_Aug2016.pdf, as of 1/1/2024.

[8] Forty-Seven Effects, "Arduino MIDI library," https://github.com/FortySevenEffects/arduino_midi_library/, as of 1/1/2024.

[9] J. Glatt, "Tick," http://midi.teragonaudio.com/tech/midispec/tick.htm, original site, now defunct, was http://www.borg.com/~jglatt. As of 1/1/2024.

[10] J. Lazzaro and J. Wawrzynek, "RTP payload format for MIDI," https://datatracker.ietf.org/doc/html/rfc6295, 2011, as of 1/1/2024.

[11] S. Jungleib, *The Complete SCI MIDI*, Sequential Circuits, Inc., 1983.

[12] S. Luke, "So you want to write a patch editor," Available at http://cs.gmu.edu, George Mason University, Tech. Rep. GMU-CS-TR-2023-1, 2018.

[13] *CC #88 High Resolution Velocity Prefix (CA-031)*, MIDI Manufacturers Association, 1996.

libremidi: a cross-platform library for real-time MIDI 1 and 2

Jean-Michaël Celerier[1]

[1]**Société des Arts Technologiques**, Montréal, QC, Canada

ABSTRACT

MIDI (Musical Instrument Digital Interface) has been a fundamental backbone for communication between digital music devices in modern music production. With the recent release of the MIDI 2 standard, available on Linux and macOS and soon on Windows, there is a now need for a robust, efficient, and easy-to-use cross-platform MIDI 2 library. This paper introduces libremidi, a cross-platform C++ library stemming from RtMidi and ModernMidi, rewritten from the ground up for real-time MIDI 2 communication. libremidi provides a simple and consistent API for managing MIDI ports, including hot-plug support, and handling MIDI events. It supports multiple platforms, including Linux (through ALSA RawMidi, ALSA Sequencer, and PipeWire), Windows, macOS, iOS, FreeBSD, and JACK on all platforms, and abstracts the underlying platform-specific MIDI APIs into a unified interface while enabling the end-user to have precise control over the back-ends. Designed for applications that require real-time MIDI communication, such as music production software, digital audio workstations, and interactive installations, libremidi's efficient and low-level API allows developers to build responsive and high-performance applications that can handle multiple MIDI inputs and outputs simultaneously. Work has also been done to approximate real-time guarantees by avoiding memory allocations altogether during input and output. This paper will provide an overview of libremidi's architecture, API, features and improvements over the current cross-platform MIDI state of the art.

1. INTRODUCTION

Communication between Digital Music Instruments (DMIs) and synthesizers, audio workstations, and more generally any kind of music-oriented tool, has been a long-standing field of work in the sound and music computing community: our tools require protocols for communication and interoperation. Published in 1983, the initial MIDI specification has been a game changer for the music and media industry, with it quickly becoming the standard data exchange protocol between DMIs[4]. While originally targeting synthesizers, computers have quickly incorporated real-time MIDI input-output features: in 1985, the Atari ST computer provided built-in MIDI ports. It sold to 2.1 million units.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Every non-trivial computing platform since then has incorporated MIDI: in particular, all the major desktop operating systems and software platform have some form of MIDI implementation. This has led to the creation of multiple cross-platform abstractions for MIDI, to enable music software authors to write software that would be able to input and output MIDI no matter the operating system. Among those, the C and C++ libraries JUCE [6], PortMidi [1] and RtMidi [7] are widely used in the media arts field and serve as base layer of many cross-platform applications and libraries in higher-level languages such as Python, LISP, SuperCollider, etc. or within frameworks such as OpenFrameworks.

As part of the ossia [2] project, a MIDI library was necessary. Originally based on RtMidi, this library has evolved over time in many ways: first by integrating ModernMidi (`github.com/ddiakopoulos/modern-midi`) a MIDI file reading and writing library and sharing most of the common code between those, for instance the in-memory representation of MIDI messages, and then refactoring and improving the library for the needs of ossia, along with updating it to contemporary practices in the C++ community and leveraging modern C++ standards where they allows tangible improvements. In particular, many safety improvements were implemented to the API, more back-ends, more configurability option, and more recently, MIDI 2 support.

This paper gives an overview of the current state of libremidi, including the ways it improves upon the current open-source cross-platform C++ libraries for MIDI communication and solves common requirements in MIDI-compatible software, usually bereft of features such as hot-plugging support. The library documentation is available at the following link: `celtera.github.io/libremidi/foreword.html`. The aim is to enable software authors to trivially capture the benefits of the modern MIDI standards, and improve the stability and reliability of their MIDI-based software.

2. CORE IMPROVEMENTS

2.1 MIDI 2 support

The library has been overhauled to support the new MIDI 2 standard by directly communicating with the relevant operating system APIs and giving to the end-user access to UMP (Universal MIDI Packets) for sending and receiving messages. Note that operating system support is barely maturing: MIDI 2 is supported by macOS since macOS 11 (November 2020), Linux since kernel 6.5 (September 2023), and still not officially supported by Windows but an experimental driver is available (`github.com/microsoft/`

MIDI) and will be distributed with Windows 11 when finished. The MIDI 2 support provided by libremidi only currently covers real-time I/O and not Standard MIDI Files 2 reading and writing.

2.2 New back-ends

A major improvement to the library is the variety of system back-ends supported. Table 1 shows the supported back-ends per-platform. As far as we know, libremidi is the only cross-platform MIDI library to support back-ends such as PipeWire and Windows MIDI Services. The only remaining major back-end to support is Android MIDI. In particular, the PipeWire back-end enables direct support for BLE MIDI on Linux.

Platform	Available backends
Linux, BSD	ALSA RawMidi **ALSA RawMidi UMP** ALSA Sequencer **ALSA Sequencer UMP** JACK PipeWire
macOS	CoreMIDI **CoreMIDI UMP** JACK
iOS	CoreMIDI **CoreMIDI UMP**
Windows	WinMM WinUWP **Windows MIDI Services** JACK
Web	WebMIDI through Emscripten

Table 1. Backends available in libremidi. The MIDI 2 backends are indicated by **bold** text, the other backends are MIDI 1. All backends support input, output and hot-plug.

2.3 Hot-plug support

A major improvement compared to PortMidi and RtMidi is support for hot-plug detection: an `observer` API allows the user to plug-in callback functions that will be called whenever a MIDI device is connected or disconnected from the computer. It allows to filter hardware and / or software devices, for APIs that provide access to cross-process communication such as JACK and PipeWire. For back-ends which do not have any hot-plugging mechanism built-in, detection is simulated with a background thread updating the list of connected MIDI devices regularly.

3. API IMPROVEMENTS

3.1 Improved reliability

Focus has been done on improving the reliability of the original RtMidi implementation, which ended up overtime into almost an entire rewrite due to fundamental reliability issues with the API, due to reliability issues with some back-ends, in particular the Windows Multimedia one.

Among other improvements:

- The entire library goes through testing with the thread, address and undefined behaviour sanitizers [8, 9].
- Callback functions are passed to the constructor to make sure that they are set before the MIDI port can be opened and streaming can start, which was an unsafe possibility and cause of data races in the original library. This also means that the callbacks will by design never miss any message as they will be set before it is possible for a MIDI port to be opened.
- Replace the handling of ports through a numerical index, by a handle object that contains all the necessary information to recover the actual port. This is in particular important in case of unplugging / re-plugging a MIDI device: in the previous design the index of the port could change after the user listed the ports and selected one. Now, the back-ends will leverage more information to try to recover which device to connect to, such as its name, or any extended unique identifier provided by the host API. The reliability of the library in case of for instance semi-faulty USB cable or socket is thus greatly improved.

3.2 Leveraging modern C++ for fun and profit

Some standard C++ types are used to harmonize the library with current C++ ecosystem norms:

- Callbacks are passed through `std::function` instead of separate (data pointer, function pointer) pairs. This allows for instance to simply use C++ lambda functions or function objects for MIDI callbacks.
- Instead of `std::vector` being used as data type for sending messages, `std::span` is used. This required setting the minimum C++ standard version to C++20, which is available on every contemporary system in 2024: continuous integration continually tests on Debian Bullseye, Bookworm, Trixie, macOS, Windows with the three mainstream C++ compilers: clang, GCC and MSVC. This allows the library to be agnostic to the data type used by its user for sending MIDI 1 messages: `std::vector`, `std::array`, raw C arrays, `boost::container::small_vector`, etc. – the only requirement is for the type to model a contiguous sequence of bytes. This removes the need for memory allocations if the end-user was using a different type than `std::vector` for internal MIDI byte storage.
- Back-ends use `std::semaphore` for cross-platform semaphore support. That is especially important for the JACK backend, now supported on macOS and Windows in addition to Linux with a single implementation.

4. CONTROL TO THE USER

A driving force in the libremidi design is to give as much control to the user as possible while still maintaining a cross-platform API for input and output management. This has been done at multiple levels in our proposed evolution of the library.

4.1 Custom configuration mechanism

A configuration mechanism for construction of MIDI input, output and observer objects was introduced. The configuration contains both a generic part common to all back-ends and an API-specific part.

Listing 1 showcases an example: how the end-user can create a CoreMIDI client with specific settings. This solves long-standing issues with multiple MIDIClient opening and closing in the same process [1].

Listing 1. Creating a MIDI device with a CoreMIDI-specific configuration, allowing to plug-in a global CoreMIDI MIDIClientRef object:

```
using namespace libremidi;

// Generic configuration
input_configuration in_config;
in_config.on_message = [] (auto msg) {
  // Process the message
};

// Specific configuration
coremidi_input_configuration in_api_config;
in_api_config.client_name = "My client";
in_api_config.context = getMyAppMIDIClientRef();

midi_in in{in_config, in_api_config};
in.open_virtual_port();
```

4.2 Removal of the implicit queue

The original RtMidi implementation allowed retrieving MIDI input messages from within the main thread, by implementing a queue (whose implementation is currently not thread-safe due to concurrent access to non-atomic variables). This feature was removed in libremidi for two reasons: first, the research field of thread-safe lock-free ring-buffers and queues is in constant evolution, with new solutions appearing in the recent years pushing the boundaries of efficiency [3, 5]. Second, we argue most professional audio and MIDI applications already carry preexisting implementations of a lock-free queue or ring-buffer, which can be trivially fed in the callback provided by the user. Removing this simplified the code logic in all back-ends. Sample code files showcasing the integration with both a custom message queue and a C++20 coroutine-based system are provided with the library.

4.3 Logging

Logging in PortMidi is done through `printf` calls, and in RtMidi through `std::cout` and `std::cerr` calls. This makes precise control of the logging in host applications impossible as one would not be able to redirect such calls done in the threaded loop of the library with full reliability. libremidi allows the end-user to provide explicit logging callbacks passed to the object constructors so that error messages can be redirected wherever fit.

[1] https://github.com/thestk/rtmidi/issues/155

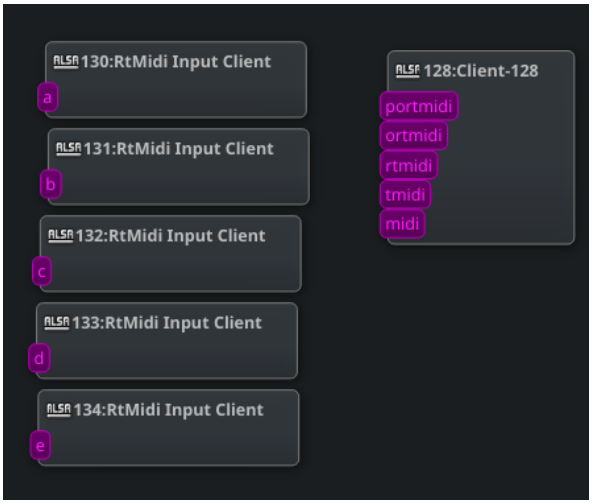


Figure 1. A patchbay with virtual MIDI input ports opened by RtMidi (left) and PortMidi (right)

4.4 Context sharing

To give context on the existing state of the art, RtMidi did not allow any sharing: each MIDI object would create a new client with a single port. Conversely, PortMidi would create all the ports in a single client. This can be seen in Fig. 1, which shows the state of the MIDI graph on a Linux system when creating five virtual ports in both these libraries. For improved flexibility, libremidi's design allows the user to pass an existing API-specific client object to enable creating ports in a preexisting, possibly shared, context: a single or multiple ALSA, JACK client, PipeWire filter, etc. Thus, the end-user gets maximum control over the context of their MIDI ports.

4.5 Poll and thread-loop management

Every API handles input and output differently. They can be split in three major categories for input: poll-driven, message-callback-driven and synchronous-callback-driven. Poll-driven (ALSA) means that the API provides a file descriptor to be polled with the UNIX `poll` API. Readiness of the file descriptor will indicate that new messages can be read. Message-callback-driven (Windows APIs, CoreMIDI) means that an operating-system-managed thread will call a user-provided callback on every new incoming MIDI message. Synchronous-callback-driven (JACK, PipeWire) means that the API calls a user-provided callback at regular interval, generally driven by the audio rate / audio buffer size.

For message-callback-driven APIs, there is no meaningful control possible: the OS handles the threading associated with the library internally. For other APIs, it may be beneficial to share the threading and polling loop across instances, or even with other operations that MIDI I/O. For

instance, a simple visualization app that does not want to bother with the risks of multi-threading may want to run the ALSA sequencer polling in an existing poll-driven application main-loop to have everything happen in a single thread. Another important capability missing from PortMidi and RtMidi is the ability to combine audio and MIDI processing in a single callback operation for instance for DAWs using APIs that support both audio and MIDI ports from the same client, such as JACK and PipeWire, and for which we want to make sure that the flow of operation on each audio scheduler tick is: *Read MIDI input → Process audio → Write MIDI output*.

libremidi's design allows for the relevant back-end to plug callback functions that when present will make the implementation refrain from creating its own thread or start its own processing through JACK / PipeWire: instead, the user will be provided with functions that they can themselves add in their audio callback processing loop, to guarantee orderly, frame-accurate synchronous operation. Examples are provided for each back-end where this feature is relevant.

4.6 Error handling

Error handling in RtMidi was done through exceptions. While in line with the C++ RAII-based object model, this can prove challenging in the audio field where it is common to require software to be compiled without exception support. Error handling across the library was thus redeveloped by leveraging the latest C++ developments on error-code-based error handling: `std::error` as proposed by Herb Sutter in [10]. We use for this an open-source prototype implementation (`github.com/charles-salvia/std_error`) that should be close to the final version expected for C++26. In particular, this implementation enables the user to react both to a given platform-specific error, for instance by retrieving an `OSStatus` on macOS or `MMERR` on Windows, viewing it as a more generic cross-platform `std::errc` POSIX error code through an equivalence class system, and getting a text string, all the while not requiring any memory allocation and being fully supported in header-only libraries and across DLLs, unlike the C++11 solution (`std::error_code`).

4.7 Advanced time-stamping

The library provides an extensive array of time-stamping options: the user can choose at run-time between:

- No time-stamping.
- Relative timestamps between successive messages. The original RtMidi behaviour.
- Absolute timestamp as provided by the OS API. A utility function has been added to enable the end-user to query the API for its current time – very often, this is the system's monotonic clock.
- Absolute timestamp as provided by the OS-provided monotonic clock. This allows to very easily have coherent time-stamping across all the parts of a media application but may lose precision compared to the API timestamp which may be able to leverage hardware MIDI time-stamping.
- Audio frame timestamp. This will for the JACK and PipeWire cases use the offset in samples as timestamp, to enable sample-accurate operation without requiring calculations from the end user.
- Finally, the user can provide a custom time-stamping callback, to leverage for instance an external clock source.

In addition, the timestamp format has been changed from double-precision to 64-bit integer with nanosecond precision to avoid subtle precision issues (except in the audio frame case where the unit of time is then the audio frame index).

4.8 Compilation options

The library has been changed so that it could optionally be used *header-only*: that is, one can compile a MIDI example with a simple invocation:

```
$ g++ \
  examples/midi2_echo.cpp \
  -Iinclude \
  -DLIBREMIDI_HEADER_ONLY=1 \
  -DLIBREMIDI_ALSA=1 \
  -std=c++20
```

The library also enables using different types than a heap-based dynamic vector, `std::vector`, for storing of MIDI 1 messages: for instance, it is possible to leverage `boost::small_vector` or even `boost::static_vector` to reduce heap allocations further, by toggling CMake flags. These data types implement respectively small-vector optimization (up to 20 bytes can be stored in the object without requiring memory allocation) or use of fixed storage to prevent dynamic memory allocations altogether. Note that for MIDI 2 support, no dynamic allocation is used in any case.

4.9 Dynamic loading of dependencies

Unlike most libraries, libremidi loads all its dependencies on Linux through `dlopen` / `dlsym`. This allows to provide a build of the library that supports all back-ends (JACK, PipeWire, ALSA) but will still work if the user's system does not provide the required library, unlike traditional dynamic linking which would fail on startup if the end-user's machine does not have all the necessary libraries. This is especially necessary for JACK and PipeWire which not everyone may be running: we want the library to be usable in the context of AppImages which can run on a wide variety of Linux distributions.

5. FROM MIDI 1 TO MIDI 2

The major advance in the library is support for the new MIDI 2 standard. We base ourselves on the June 2023 MIDI 2 standard documents [2]. This section gives an overview of the MIDI 2 standard, and details how it is implemented in libremidi.

[2] `https://midi.org/details-about-midi-2-0-midi-ci-profiles-and-property-exchange-updated-june-2023`

5.1 A MIDI 2 primer

MIDI 2 is a major advance in the MIDI protocol. It is currently still evolving, with the latest changes to the standard documents being dated from June 2023.

Multiple specifications define what MIDI 2 exactly is:

> MIDI 2.0 builds on the core principles, architecture, and semantics of MIDI 1.0. Primary features are auto-configuration, enabled by bidirectional connections enabling devices to discover details about other connected devices, and an expanded data format for higher resolution with extensibility to define many new messages in the future.

The core elements of MIDI 2 are: MIDI Capability Inquiry (MIDI-CI) discovery and the UMP message format. To be a compliant MIDI 2 device, one has to implement at least one of these two. In particular, MIDI-CI works over a MIDI 1 transport through SysEx messages and does not require UMP support.

5.2 Core differences with MIDI 1

For artists, the most immediate benefits will be, in short, "more data": pitch bend, control change and polyphonic aftertouch now carry 32 bits of information instead of MIDI 1's 7 bits ; note messages 16 bits of velocity and 6 bits of custom attribute data ; instead of only 16 channels, a MIDI 2 link can carry 16 groups of 16 channels each ; new controllers are introduced ; SysEx messages can now use 8-bit bytes instead of 7-bit in MIDI 1. For implementers, the low-level protocol is much more complex with multiple round-trips to define the capabilities of MIDI 2 devices, which are supposed to be retro-compatible with MIDI 1.

For library users such as the target audience of this paper, many design choices in MIDI 2 also greatly simplify usage of MIDI 2: we recommend that over time every desktop computer user migrates to MIDI 2 as an end-user API. In particular, the three major desktop operating systems, Linux, macOS, and Windows, provide automatic translation of MIDI 1 devices to the MIDI 2 API.

In particular, MIDI 2 messages are now in a new, fixed-size format, unlike MIDI 1 messages which could be of arbitrary-length due to the SysEx format. The Universal Midi Packet (UMP) can be 32, 64, 96 or 128 bits, generally organized as four unsigned 32-bit values. This obviates the need for dynamic memory allocation which was often necessary with SysEx support in MIDI 1 APIs, and is a better fit to the current CPUs, with generally 64-bit registers and 64-byte L1 cache lines.

Another feature, necessary in the MIDI 1 times and made irrelevant with current hardware performance, is the running status which is not part of MIDI 2 anymore: every message now carries the entire information needed to interpret it. Finally, this UMP message format carries more space for future extensions to the protocol: many are already in the works by the MIDI association.

Besides the improved resolution, MIDI 2 also provides a complete protocol for bidirectional information exchange between devices, MIDI-CI. This allows two MIDI-CI-compatible devices to interoperate more closely: a synthesizer can now for instance report its controls to a MIDI controller. MIDI-CI communication follows rules established in the MIDI standard documents.

6. RESULTS AND DISCUSSION

The library has already seen usage in the media ecosystem: it has been validated by years of use in the ossia score software sequencer and has also been integrated in multiple open-source projects, for instance OBS Studio's MIDI plugin.

6.1 Porting to libremidi

Multiple examples are provided by the library, showing different degrees of integration. As a starting point, Listing 2 shows how one may implement a basic MIDI 2 echo between the first two detected devices.

Listing 2. Minimal MIDI 2 echo example

```
#include <libremidi/libremidi.hpp>
#include <iostream>

int main() try {
  using namespace libremidi;
  namespace lm2 = libremidi::midi2;

  /// The observer object enumerates available
  /// inputs and outputs:
  observer obs{{}
    , lm2::observer_default_configuration()};
  auto pi = obs.get_input_ports();
  auto po = obs.get_output_ports();
  if (pi.empty() || po.empty())
    throw std::runtime_error("No MIDI in / out
        available");

  /// Create a MIDI 2 out object:
  midi_out midiout{{}
    , lm2::out_default_configuration()};

  // Open the midi output object with a given
  // handle obtained from the observer
  if (auto err = midiout.open_port(po[0]);
      err != stdx::error{}) {
     // error-handling decisions are deferred
     // to the user of the library.
     // Here for simplicity we throw.
    err.throw_exception();
  }

  /// Create a MIDI 2 input object:
  // An UMP-friendly callback
  auto on_ump =
    [&](const libremidi::ump& message) {
    // Note: one may want to lock the
    // midiout object with a mutex if
    // it was to be used at the same time
    if (midiout.is_port_connected())
      midiout.send_ump(message);
  };
  midi_in midiin{{.on_message = on_ump}
    , lm2::in_default_configuration()};

  // pi[0] and po[0] are distinct types:
  // one cannot mistakenly open an input
  // with an output handle
  if (auto err = midiin.open_port(pi[0]);
      err != stdx::error{})
    err.throw_exception();

  // Wait until we exit
  char input; std::cin.get(input);
}
catch (const std::exception& error) {
        std::cerr << error.what() << std::endl;
        return EXIT_FAILURE;
}
```

6.2 Performance results

Summary performance benchmarking of a MIDI echo implementation between libremidi and RtMidi has been done. A core point though is that RtMidi does not allow accurate benchmarking: the RtMIDI output does not inform the user of an error if a message could not be written (for instance if the operating system's internal MIDI buffer is full, its `sendMessage` function can fail silently, e.g. in the JACK back-end). In contrast, libremidi allows to take action and count the number of actually successfully transmitted messages. With this taken into account, our sample benchmark run (sending 1 000 000 messages as fast as possible in a round-trip fashion on Linux with the ALSA Sequencer back-end on a Core i7-10750H CPU, five runs, `-Ofast -flto`) gives an average of 720ms (best: 700.3) with libremidi, and 816 (best: 787.9) with RtMidi. That said, most of the MIDI performance comes from the underlying back-end implementation and both libraries have very little overhead: 88.5 percent of the execution time of our `send_message` implementation is spent inside the ALSA API.

6.3 MIDI-CI support ?

In short, from our prolonged study of the MIDI 2 standard and implementation experience since the first MIDI 2 document release in 2020, our conclusion is that a library such as libremidi does not operate at the abstraction level at which MIDI-CI is useful, which is specific to each application. On the other hand, providing UMP support is the right fit: libraries defining a specific MIDI 2 state machine can then just use libremidi for the cross-platform input-output aspect and feed MIDI-CI messages to an appropriate MIDI 2 data processing library. Multiple libraries implement C and C++ API support for high-level MIDI 2 messages, such as creating UMP packets matching a specific step of a MIDI-CI communication from simple function calls. For the sake of ease-of-use, libremidi has been made compatible with the two major implementations: `ni-midi2`[3] developed by Native Instruments and `cmidi2`[4] developed by Atsushi Eno. In particular, one can directly send `ni-midi2` data types to libremidi outputs, and the libremidi message type can be converted to the `ni-midi2` UMP data type automatically. Examples in the codebase show how to initiate a basic MIDI-CI communication by combining libremidi and `ni-midi2`, only tested so far with the MIDI 2 Workbench software provided by the MIDI association due to the lack of MIDI 2 hardware available.

6.4 UMP, a friendly format

Every desktop platform provide automatic MIDI 1 support with their UMP APIs: that is, one can just use the MIDI 2 UMP APIs, forget about MIDI 1 entirely and get access to the entirety of the MIDI 1 and MIDI 2 devices, at the cost of restricting the applications to recent operating system versions (macOS 11+, Linux 6.5+, and yet unreleased Windows versions). Due to the major advantages of the UMP format, we heartily recommend a complete migration of MIDI support in media applications to the UMP format. The only drawback is that some back-ends do not provide MIDI 2 support yet: JACK, PipeWire, WebMIDI. Both `cmidi2` and `ni-midi2` provide efficient implementations of the conversion from MIDI 1 messages to MIDI 2 UMPs. libremidi leverages this for the MIDI output: all the back-ends supported by the library can receive UMPs including the MIDI 1 ones. Conversion of MIDI 1 input to UMP is a work-in-progress.

[3] `github.com/midi2-dev/ni-midi2`
[4] `github.com/atsushieno/cmidi2`

7. CONCLUSIONS

We introduce libremidi, a modern C++ take on the perennial problem of cross-platform MIDI I/O library. Based on preexisting code, it evolved to enable support of the more recent MIDI 2 standard, and provides improved safety and reliability over existing C & C++ MIDI libraries. Future directions will mainly be about providing more back-ends, for instance for Android and BSDs, and ensuring stability of the library now that the necessary features are there.

8. REFERENCES

[1] R Bencina, P Burk, and R Dannenberg. *portmidi-Platform Independent Library for MIDI*. 2007. URL: `https://github.com/PortMidi/portmidi`.

[2] Jean-Michaël Celerier et al. "OSSIA: Towards a Unified Interface for Scoring Time and Interaction". In: *Proceedings of the International Conference on Technologies for Music Notation and Representation (TENOR)*. Paris, France, 2015.

[3] Reginald Austin Frank. "Designing a High Throughput Bounded Multi-Producer, Multi-Consumer Queue". PhD thesis. 2021.

[4] Gareth Loy. "Musicians make a standard: The MIDI phenomenon". In: *Computer Music Journal* 9.4 (1985), pp. 8–26.

[5] Gal Milman-Sela et al. "BQ: a lock-free queue with batching". In: *ACM Transactions on Parallel Computing* 9.1 (2022), pp. 1–49.

[6] Martin Robinson. *Getting started with JUCE*. Packt Publishing Ltd, 2013. DOI: `10.4324/9780429455971-4`.

[7] Gary P. Scavone and Perry R. Cook. "RtMidi, RtAudio, and a synthesis toolkit (STK) update". In: *Synthesis* (2004).

[8] Konstantin Serebryany and Timur Iskhodzhanov. "ThreadSanitizer: data race detection in practice". In: *Proceedings of the workshop on binary instrumentation and applications*. 2009, pp. 62–71.

[9] Konstantin Serebryany et al. "AddressSanitizer: A fast address sanity checker". In: *2012 USENIX annual technical conference (USENIX ATC 12)*. 2012, pp. 309–318.

[10] Herb Sutter. "Zero-overhead deterministic exceptions: Throwing values". In: *C++ open-std proposal P0709* 2 (2019), p. 10.

PROGRAMMING FPGA PLATFORMS FOR REAL-TIME AUDIO SIGNAL PROCESSING IN C++

Pierre COCHARD (pierre.cochard@insa-lyon.fr) [1], **Maxime POPOFF**[1], **Romain MICHON**[1], and **Tanguy RISSET**[1]

[1]**INSA Lyon**, **Inria**, CITI, EA3720, 69621 Villeurbanne, France

ABSTRACT

Syfala is a toolchain aiming to facilitate the development of real-time audio Digital Signal Processing (DSP) applications on Field-Programmable Gate Array (FPGA) platforms. Since its introduction in 2021, the project has been relying on the FAUST programming language as its primary programming interface. The resulting C++ code generated by the FAUST compiler can be easily translated into Hardware Description Language (HDL) code using High Level Synthesis (HLS), thereby facilitating integration into FPGA designs. However, FPGA platforms may in some cases require very specific tuning and optimizations in order to reach their full potential, which FAUST can sometimes fail to provide, because of its platform-agnostic nature.

In this technical paper, we present a new way of using Syfala with C++ and HLS. This potentially allows for a better control over parallelization and pipelining as well as for potentially more optimized and efficient code for AMD/Xilinx FPGA platforms. This method can greatly facilitate the implementation of complex DSP algorithms on FPGAs for C++ programmers. Finally, we highlight the performance gains obtained by using this methodology with respect to the original Syfala toolchain starting from FAUST programs.

1. INTRODUCTION

Field-Programmable Gate Array (FPGA) platforms offer unmatched performances and features in the context of real-time audio Digital Signal Processing (DSP). With the exception of Graphics Processing Units (GPUs) where comparable results can be achieved [1,2], highly parallelizable algorithms can be run more efficiently on an FPGA than on most other computing platforms (i.e., Central Processing Units, etc.). FPGAs don't need buffering and can run audio DSP algorithms at a very high audio sampling rate (in the megahertz range), providing extremely low latency [3]. They can also be interfaced with large numbers of low-level hardware components (such as audio codecs) through their numerous GPIOs.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

FPGAs are very difficult to program for a software programmer. However, audio DSP programs can be expressed using dedicated languages (i.e., FAUST [4]) and might be easier to compile on FPGAs than general purpose programs because of their limited scope. This technical paper has been written as a tutorial to help C++ audio programmers to efficiently compile their programs down to FPGA. It takes advantage of the methodology that emerged from the Syfala toolchain which compiles FAUST programs on FPGA [3,5]. Syfala (which is open source [1]) is now able to compile C++ audio programs on FPGA using AMD/Xilinx High Level Synthesis (HLS) tools. In this paper, we explain the concepts that software programmers must know to master HLS in order to compile efficient C++ audio programs on FPGA.

The next section presents a review of the use of FPGAs in the audio industry as well as in academia. It also describes how audio DSP programs should be compiled for FPGA. Section 3 presents the Syfala toolchain and Sections 4-5 explain how to adapt existing C++ programs. Finally, Section 6 presents some performance results obtained by this new FPGA compilation flow.

2. AUDIO ON FPGA: A STATE OF THE ART

FPGAs can be found in a broad range of audio products: Dante audio interfaces,[2] digital mixers (e.g., Midas M32[3]), audio processing modules (e.g., Antelope Audio Synergy Core[4]), sound synthesizers (e.g., Novation Summit Keybord[5]), etc. Similarly, they've been used in academia for various applications, mostly targeting computational performance [6–8] (to name a few).

2.1 Language and Architecture for audio DSP

The advantages offered by FPGAs come at the cost of ease of programming. Indeed, FPGAs are considered as "hardware" and are low-level in nature, making them extremely hard to program by non-specialized engineers. They require the use of Hardware Description Languages such as VHDL and Verilog, which work in a very different way than programming languages aiming software and that are traditionally used for real-time audio DSP such as C/C++.

[1] https://github.com/inria-emeraude/syfala/ – All URLs in this paper were verified on May 21, 2024.
[2] https://www.audinate.com/
[3] https://www.midasconsoles.com/
[4] https://en.antelopeaudio.com/
[5] https://novationmusic.com/

Most FPGA manufacturers (i.e., Intel/Altera and AMD/Xilinx) now provide High-Level Synthesis (HLS) tools allowing for the programming of FPGAs using C or C++ [9, 10]. Since programs written in these languages are often not optimized to be run on an FPGA, directives and pragmas can be used to provide additional information to help the HLS tool make the right decisions in terms of pipelining, parallelization, etc.

Audio DSP implies a broad range of constraints when it comes to real-time implementation (e.g., buffering, control vs. audio rate, interfacing with audio drivers, etc.). These constraints – despite the potential use of HLS – add up to the aforementioned difficulties related to programming FPGA platforms. As far as we know, very few projects (not to say only one [11]) used HLS on FPGA in the context of real-time audio DSP as it is done in Syfala.

Since the inception of digital audio programs, specialized languages and architectures tailored for audio Digital Signal Processing (DSP) have been proposed. Audio signal processing often demands significant computational power, implementing mathematical concepts like FIR and IIR filters, FFT, etc., at a high rate (i.e., audio sampling rate). It's important to note that a characteristic of audio signal processing is that some computations are performed for each sample, while others, like those used for audio control, can be scheduled at a lower "control rate".

In the early days of digital audio, dedicated DSP units were employed in commercial systems. This trend persists today, exemplified by the continued use of specialized DSP units like Analog Devices SHARC DSPs. However, there is a growing trend in utilizing regular multi-core CPUs for audio purposes, as seen with ARM processors in Teensy [6] microcontrollers.

The selection of hardware for implementing audio signal processing is guided by two key metrics: computational power and computation latency. Computational power depends on the operations performed for each sample at sample rate; the more demanding the computations, the more powerful the hardware needs to be. Dedicated DSPs, GPUs, or FPGAs may be employed if standard CPUs are not sufficiently powerful. As for latency, it is often associated with the size of the sample buffer needed to perform the DSP computation. The nature of the Von Neumann processor architecture (i.e., a CPU connected to memory) necessitates buffering audio samples to circumvent memory bottlenecks. A typical size for an audio buffer in processing systems is 256 samples. At a sampling rate of 48kHz, a 256-sample latency corresponds to approximately 5 ms. FPGA technologies can can help in providing much smaller latency. For instance a latency of $11\mu s$ (analog to analog) was reached in [3].

An important decision in the design of audio signal processing involves selecting the programming language to express these algorithms. Numerous computer languages and environments have been dedicated to audio programs, with today's most popular choices including Max/MSP, PureData, and Faust [4]. Conversely, many audio developments, particularly in the industry, use general-purpose languages, mainly C/C++, but also Python, Matlab, or Java.

The Syfala toolchain allows us to map Faust programs onto FPGAs [3, 5], leveraging C++ code generated by the Faust compiler. It seems that many programmers express a desire to program FPGAs directly in C++ since it serves as their native audio programming language. In the following sections, we present guidelines for writing C++ code in a manner that facilitates efficient mapping onto FPGAs using Syfala.

FPGAs present a programming model distinct from regular CPUs, as they can be regarded as "programmable circuits" in contrast to non-programmable digital circuits like ASICs. An FPGA is composed of many small *configurable* blocks, called Look-Up Tables (LUTs), capable of executing any logic function with 4 to 6 boolean inputs and storing the result in a 1-bit register. These LUTs are utilized to implement diverse digital circuits. Additionally, FPGAs incorporate various components such as embedded multipliers, known as *DSPs*, that facilitate the implementation, or *BlockRAMs* which can store several kilobytes of data.

Over the past decade, FPGAs have incorporated *processing systems*, which are system-on-chip entities featuring complete processors and associated peripheral drivers. For example, the Digilent Zybo Z20 development board includes a dual-core ARM processor capable of operating at 600MHz. This processing system supports running a complete operating system and facilitates communication with the hardware design mapped onto the FPGA LUTs. This processing system is used to offload the hardware component from computationally less time-sensitive tasks.

2.2 Mapping audio DSP on FPGAs

The process of mapping an audio application onto an FPGA involves making crucial design decisions, extensively discussed in [5]. Typically, sample-rate computations are implemented on the FPGA LUTs, while control rate computations are executed on the processing system. Another important choice is the buffering of samples; one can choose to compute one sample at a time (one-sample strategy) or a buffer of samples at a time (multi-sample strategy). Fig. 1 illustrates a generic architecture for an audio application running on an FPGA.

The hardware/software partitioning is shown on Fig. 1. It distinguishes the control-rate and the sample-rate computations. The control part runs on the processing system (refered to as `app.elf` on Fig. 1). It is itself interfaced with the FPGA board DDR memory and also to the host computer (or any computer on the network) in order to provide a way to control the audio DSP computations. The sample-rate computation are implemented on the FPGA (refered to as *DSP IP* on Fig. 1). This represents the core computation of the DSP kernel. It is interfaced with the FPGA board DDR memory, but it can also communicate directly with the `app.elf` application (i.e., the DSP IP is usually seen as a peripheral driver on the processing system). Finally, the IP is interfaced with one or several audio codecs via the I^2S protocol.

The next section explains how the system presented on

[6] https://www.pjrc.com/

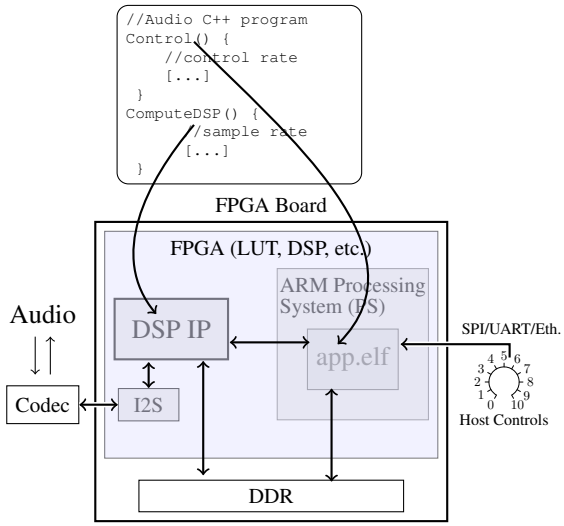


Figure 1. Overview of the deployement of audio programs on FPGAs.

Fig. 1 has been implemented in the Syfala toolchain, which compiles FAUST programs down to FPGA.

3. FROM FAUST TO FPGA: THE SYFALA TOOLCHAIN

Syfala was introduced in [3, 5]. It is a tool to facilitate the development of real-time audio DSP applications on FPGAs using the FAUST programming language [4]. Programs written in FAUST can be exported to AMD/Xilinx FPGAs on Digilent boards such as the Zybo Z7 or the Genesys. [7]

3.1 Using FAUST as an Input to Syfala

Among the features provided by Syfala, one can cite the following ones:

- Many kinds of audio codecs are supported;
- The DSP IP can be used as a hardware accelerator on embedded Linux running on the FPGA board [12];
- SPI [3], MIDI, HTTP, and OSC control are available on the processing system [12];
- Ethernet can stream audio in and out from the FPGA [13];
- It is compatible with dedicated multichannel audio interfaces providing unique specific features such as low latency and low-cost spatial audio.

The "traditional" way of using the Syfala toolchain is to call the FAUST compiler on a `.dsp` file (i.e., a FAUST program) in order to generate its C++ code equivalent, which is then going to be translated into Hardware Description Language (HDL) code using High Level Synthesis tools.

The resulting code describes the DSP computations as a flow (or circuit) of hardware-level logic operations and interconnections, forming an ensemble that we call an IP (for Intellectual Property). In this context, an IP can be viewed as a higher-level 'block' abstraction that can be connected to different other blocks through a variable number of input and output ports, as shown on Fig. 2.

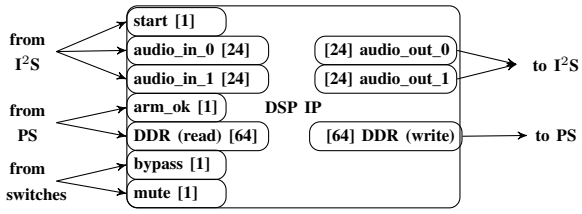


Figure 2. A logic view of the DSP IP as generated in the Syfala compilation toolchain.

When running, the data-flow design can be described as follows: audio data will be sent from the codec(s) to the I^2S at a predetermined rate (called audio-rate or sample-rate), and then transmitted to the DSP IP, alongside a 'start' signal, which will enable the computations. Once these computations are done, the output samples are sent back to the I^2S, and in turn to the audio codec(s), bit-by-bit.

Since this is all done at a hardware level, there is no additional latency introduced in the system (from, for instance, an operating system or any other hardware/software component), which makes real-time sample-by-sample processing possible in this context. The main requirement is that the computations are still done within the time limits set by the audio clock, otherwise making the flow discontinuous and hence degrading the resulting audio signal.

In order to guarantee that the latency stays below this limit, FPGAs can rely on their great parallelization potential, which on the other hand will be limited by the available on-board logic resources. Indeed, since multiple signals are processed simultaneously on parallel branches, with the same chain of operations, it also means that the logic operators used on the FPGA are duplicated instead of being sequentially re-used. This can in some cases lead quite rapidly to resource saturation.

Therefore, when programming an FPGA, having control over this latency/resource trade-off is key. The programming language that is used should in this context provide flexible ways and tools to reach optimal balance for a given program.

3.2 Using C++ as Input to Syfala

While FAUST is undoubtedly a nice and easy way to create complex and fully-controllable audio DSP IPs on FPGA, in some cases – where, for instance, balancing resource utilization and computation latency becomes a critical issue – bypassing FAUST and programming the IP directly in C++ can become a more suitable solution.

The C++ code generated by the FAUST compiler is a translation of the data-flow graph representation of its input program. Due to the compiler inherent structure, the code translation will try to express the maximum level of parallelization that its graph naturally possesses. While the FAUST compiler has a few options that allow us to tune its

[7] `https://digilent.com/reference/programmable-logic/zybo-z7/`

code-generation process, it still in essence stays platform-agnostic, and will not – at least not for the time being – make any FPGA-specific optimizations. Furthermore, since the compiler already carries out several transformation and optimization passes, the resulting code is difficult to apprehend and optimize manually.

On the other hand – since Syfala is already relying on HLS – programming the DSP IP directly in C++ is entirely possible, as it does not introduce any overhead and very little change to the rest of the compilation flow. While being more time consuming than writing the equivalent FAUST program, writing directly in C++ offers a few advantages that are not negligible:

- better control over the generated HDL code output (lower-level language);
- more potential for micro-optimizations and balance control between FPGA resource utilization and computation latency, through the use of various directives and pragmas. [8]

On the other hand, this approach requires to be familiar with the Vitis HLS programming environment and guidelines, which includes, among other things:

- use of a limited version of C++ (up to C++14), as well as a few strict rules on the use of the language's features (such as types, data structures, casts, etc.);
- implement complex HLS interfaces, using pragmas, with a potentially steep learning curve;
- handle manually data exchange with ARM executables through Memory bus interfaces.

4. DSP IP GENERATION FROM C++

In practice, writing a custom C++ DSP IP implies the same steps as writing a regular C++ program: all the code can be written in one or multiple C/C++ source or header files, and it should be C++ 14 compliant (meaning it should also compile using a regular C++ compiler) and have an 'entry point' function (called a 'top-level function' in Vitis HLS), which we will describe in the following sub-sections.

4.1 Top-Level Function Interface

The top-level function of a custom C++ IP can be seen as the equivalent of the 'main' function in a regular C or C++ program, and its arguments should be considered as the declaration of the IP's input and output ports, with:

- non-pointer arguments representing input (read) ports only;
- pointer or array arguments representing input or output ports (read, write, or both).

The listing in Fig. 3 shows a basic example of a DSP IP top-level function, made out of two audio input ports,

```
void syfala (
        sy_ap_int audio_in[2],
        sy_ap_int audio_out[2],
           int arm_ok,
        float* DDR,
          bool bypass,
          bool mute
) {
#pragma HLS array_partition variable=audio_in
    type=complete
#pragma HLS array_partition variable=audio_out
    type=complete
#pragma HLS INTERFACE s_axilite port=arm_ok
#pragma HLS INTERFACE m_axi port=DDR
    latency=30
[...]
```

Figure 3. Example of a kernel DSP top-level function used to generate the DSP IP with Vitis HLS.

two audio output ports, two additional '*bypass*' and '*mute*' input ports, as well as a few memory bus interfaces (the *arm_ok*, and *DDR* ports). As one can see, all of these interfaces are specified with a set of HLS-specific *pragmas*, which can themselves be configured with a few additional parameters. These top-level function arguments are explained below.

4.1.1 Audio Input/Output Signal Types

In standard Syfala designs, audio signals are conveyed from and to the audio codecs as streams of 24-bits integers, by default (16 or 32 bits can be used as well). The top-level argument type `sy_ap_int`, which is used for audio input and output ports, is a specialization of a generic fixed-point integer type defined by Vitis HLS in its core libraries. Its width can be changed using the `-sample-width` flag in the syfala command line interface, but it cannot be changed to single or double precision floating point types.

Since audio DSP programs are usually processing float or double-based signals, a few convenience functions (`ioreadf` and `iowritef`) and types for making the proper conversions have been added in the Syfala API headers, which can be directly included in the target C++ file. From these header files, it is also possible to access some useful compile-time data, such as the project's sample rate, sample width, or targeted board model.

4.1.2 PS-PL Memory Interfaces

The two following top-level function arguments (*arm_ok*, *DDR*), and their respective pragma settings, show the use of *Advanced eXtensible Interface* (AXI4) memory buses, in order to create and enable shared memory areas between the DSP IP (which dwells on the Programmable Logic, or PL) and the Processing System (PS). This will allow us, among other things, to share data between the DSP IP and an ARM CPU-based executable, both running simultaneously on the same System on Chip (SoC).

In this context, data can either be shared using the DDR memory controller (through an AXI4 bus), or smaller intermediate PL Block RAMs (BRAM), using a lighter subset of the protocol (AXI4-Lite). In Syfala, these interfaces are implemented by default on standard designs, and may be used for multiple purposes (see Section 5.1). The `arm_ok`

[8] Directives and Pragmas are provided by Vitis HLS to *tune* parallelization and allow for a balance between FPGA resource utilization and computation latency of the DSP IP design.

argument is used in this context as a synchronization point between the DSP IP and the ARM executable while the DDR pointer is used to store and retrieve floating-point data in DDR memory.

4.2 DSP Kernel Code

Inside the top-level function, the code dedicated to signal processing can be written like standard C/C++ DSP code. The only requirement is to switch back and forth between *float* and *sy_ap_int* in order to read the input signal arguments, and write back to the outputs, which can be easily done by taking advantage of some of the API functions, as shown below.

```
for (int n = 0; n < 2; ++n) {
    // Read audio inputs
    float s = Syfala::HLS::ioreadf(audio_in[n]);
    // Do something with s
    s *= 0.5s;
    // Write audio outputs
    Syfala::HLS::iowritef(s, audio_out[n]);
}
```

4.2.1 Sample-Block Configuration

By default, Syfala implements the *one-sample* stategy. It is nonetheless possible to configure the DSP IP to process a buffer of samples instead, for potential optimization purposes. Using the multi-sample strategy may succeed in decreasing the *latency per sample* performance of the DSP IP, taking advantage of the pipelining optimizations offered by Vitis HLS, which might be useful in some cases.

The multi-sample strategy is simply activated by adding the *–multisample <N>* flag to a syfala command. However, the DSP IP code also needs to be adapted, since it requires replacing the single-valued input and output ports of the top-level function by *first-in-first-out* (FIFO) arrays, which should be declared as C multidimensional arrays in the HLS context, as shown below.

```
void syfala (
    sy_ap_int audio_in[2][SYFALA_NSAMPLES],
    sy_ap_int audio_out[2][SYFALA_NSAMPLES],
    [...]
){
#pragma HLS INTERFACE ap_fifo port=audio_in
#pragma HLS INTERFACE ap_fifo port=audio_out
#pragma HLS array_partition variable=audio_in
#pragma HLS array_partition variable=audio_out
```

4.2.2 Internal Loops Pragmas

Finally, code optimization in the HLS environment is usually done using various sets of pragmas, which can be viewed as higher-level hints or instructions given to the compiler to process a specific part of the code in a certain way. These pragma declarations usually target an individual variable, function or loop, and can often be tuned with various parameters.

The complete mastering of all Vitis HLS pragmas is out of the scope of this paper, but one needs to know that many adjustments can be done with *loop unrolling* and *loop pipelining*. Loop unrolling (`HLS unroll` pragma as below) is used to to increase parallelism by using additional resources. If a loop is not unrolled, it will be implemented sequentially (i.e., no additional resources) and Vitis HLS will try to pipeline the loop, i.e., overlap successive iterations without using additional resources.

In the code below, 4 iterations of the first loop will be executed in parallel and the second loop will be pipelined. Increasing the unroll factor will reduce the overall latency and increase resource usage:

```
for (int i = 0; i < NMODES; i++){
    #pragma HLS unroll factor=4
    for (int n = 0; n < SYFALA_NSAMPLES; ++n) {
        #pragma HLS pipeline
            [....]
        }
    }
```

4.3 Additional HLS Programming Environment Tools

Depending on the actual code and machine it is running on, using the Syfala toolchain to build a project based on a FAUST or a C++ file can take quite some time: one can expect a total build duration somewhere between 15 and 60 minutes. Hence, before running hardware synthesis, the two following points have to be checked, at least:

1. the DSP IP code is valid (it produces the outputs it is supposed to, from the inputs that it is given);
2. its implementation fits on the targeted platform, both in terms of FPGA resources and computation latency.

Fortunately, Vitis HLS offers tools that can make this kind of verification much faster (i.e., few seconds): C simulation (CSIM) and synthesis reports, which are also both well-integrated to the Syfala toolchain workflow.

4.3.1 Verifying Code With C Simulation (CSIM)

C simulation is an important Vitis HLS feature that allows us to test a C-written custom IP without having to get through the full synthesis process. Vitis HLS guarantees (with a few exceptions) that the outputs produced during simulation are going to be the same as what would happen in a "real" context of execution (implemented as hardware).

In Syfala, this simulation process can be directly ran from the command-line interface, and the Syfala documentation website [9] offers in its tutorials a few templates and guidelines on how to write a custom CSIM program.

4.3.2 Monitoring Latency & Resource Usage

Once it is established that the DSP IP code is valid, the other essential information to obtain is whether the design fits on the targeted board, both in terms of FPGA resource usage and latency. For the latter, it is also important to make sure that the DSP computations are within the time limits set by the FPGA clock rate, as well as the project's sample-rate constant. For instance, the DSP IP computations should last less then 2604 FPGA clock cycles for the computation of a single sample, for a sample rate of 48kHz and a FPGA clock rate of 125MHz.

[9] https://inria-emeraude.github.io/syfala/

After running the high-level synthesis step, the Syfala command-line interface automatically displays in the terminal an overview of Vitis HLS resources and latency estimates for the current program. If a more accurate report is needed, the *–accurate-use* flag can also be added to the command, which runs the implementation step of the design on the targeted platform. The *–accurate-use* flag usually adds a few more minutes to the HLS process to complete.

5. ARM EXECUTABLE INTERFACES

In some cases, it may be necessary to provide Syfala with a *custom* ARM executable in order to address specific needs, such as:

1. Using memory buses to share custom data and computations, as mentioned in Section 2.1;
2. Overriding the default configuration of the platform's peripherals (when using new external codecs);
3. Adding external software components, or implementing custom control interfaces, using for instance the Musical Instrument Digital Interface (MIDI) or Open Sound Control (OSC) protocols.

Custom ARM executables can easily be interfaced with the DSP IP and the system peripherals through a set of modules provided by the Syfala ARM API, which have separate implementations in baremetal and embedded-Linux build contexts.

An example of code for a minimal ARM executable, in embedded Linux context, can be seen in Fig. 4, in which a few different API modules are included, configured and initialized, all within the scope of the `main` function. This custom program can be added as a replacement to the default one provided by Syfala using the `-arm-target <file.cpp>` option in the command-line interface.

```
#include <syfala/arm/audio.hpp>
#include <syfala/arm/gpio.hpp>
#include <syfala/arm/uart.hpp>
#include <syfala/arm/dsp.hpp>

using namespace Syfala::ARM;
int main()
{
    XDSP dsp_ip;
    UART::data udata;

    GPIO::initialize();
    UART::initialize(udata);
    Audio::initialize();
    DSP::initialize(dsp_ip);
    DSP::set_arm_ok(&dsp_ip, true);
    Status::ok("Application ready, now running");
    // Main event loop:
    while (...) {
        // ...
    }
    return 0;
}
```

Figure 4. Example of minimal control program running on the ARM processor (i.e., `arm.elf` on Fig. 1).

The Syfala ARM API documentation, which describes in details all of the available modules and their interface, can be browsed on the Syfala documentation website. [10] Since the DSP and Memory modules constitute special cases insofar as they can be used to ensure interoperability between the ARM and the FPGA, we will also describe in the following subsections how they work and the benefits they are able to provide.

5.1 ARM/FPGA Interoperability

Sharing data and processing between PS and PL can prove essential for making the load lighter on the DSP IP. Hence the ARM executable can be used with specific strategies in mind, such as:

- The initialization of data arrays (wavetables) or constant expressions, by the ARM executable;
- Storing long delay-lines in DDR memory, instead of BRAM;
- Computing control-rate signals.

In order to implement these different strategies, sharing data between the DSP IP and the ARM executable is key, and can be easily achieved by leveraging the Advanced Microcontroller Bus Architecture (AMBA) interfaces set up on the System on Chip, specifically the AXI4 bus systems, which are natively supported in Vitis HLS.

5.1.1 AXI4-Lite Interfaces

The AXI4-Lite bus protocol is the quickest and simplest way to setup data exchange between a custom-made IP and an ARM executable. When adding an AXI4-Lite-registered argument to the top-level function of a custom C++ IP, Vitis HLS will automatically generate its matching 'driver' getter and setter functions in a C header/-source file during synthesis. An example of the generated `Set/Get` function prototypes for the `arm_ok` argument seen in Fig. 3 is displayed below. The registered argument (`arm_ok` here) can either hold a single value, or an array of multiple values.

```
// xdsp.h
typedef struct {
    u64 Control_baseAddress;
    u32 IsReady;
} XDSP;

void XDSP_Set_arm_ok(XDSP *InstancePtr, u32 Data);
u32 XDSP_Get_arm_ok(XDSP *InstancePtr);
```

From the ARM executable point of view, the generated functions and their common C-struct handle (such as the one above) can then be accessed and called. Syfala will then take care of automatically including and linking the generated C sources when building the executable.

5.1.2 AXI4 Memory Interfaces

In order to read and write through the AXI4 bus, the process is somewhat similar, since it relies on the same Vitis HLS function-generation mechanism, but the data that is passed to the generated functions now consists in pointers:

[10] `https://inria-emeraude.github.io/syfala/`

```
// xdsp.h
u64 XDSP_Get_DDR(XDSP *InstancePtr);
```

For convenience, the Syfala ARM Memory module can be called to take care internally of passing valid and exclusive DDR memory addresses to the DSP IP, which can then be retrieved in the *DDR* pointer argument in the HLS code (as seen in Fig. 3).

```
[...]

#include <syfala/arm/memory.hpp>
using namespace Syfala::ARM;

#define WAVETABLE_LEN 16384

static void
initialize_wavetable(Memory::data& m) {
    int w = WAVETABLE_LEN;
    for (int n = 0; n < w; ++n) {
         mem.f[n] = sin((float)(n)/w * M_PI * 2);
    }
}

int main() {
    [...]
    Memory::data mem;
    Memory::initialize(dsp_ip, mem, 0, WAVETABLE_LEN);
    initialize_wavetable(mem);
    [...]
    Status::ok("Application ready, now running");
    while (...) {}
    return 0;
}
```

The code above shows an example of using the Syfala Memory module in order to setup a basic sinewave table on a custom ARM executable. Since the computations for the initialization of the wavetable are only made once, it is best to let the CPU handle them, instead of using FPGA resources. The resulting data is then shared internally with the DSP IP through the AXI4 bus, and handled in the HLS code using the *DDR* pointer.

5.1.3 AXI4 vs. AXI4-Lite

Although AXI4-Lite is usually faster when sharing a restricted amount of data, due to that fact that it does not implement all of the AXI4 bus features, it can also potentially lead to bottlenecks when this amount significantly increases. Unlike AXI4, AXI4-Lite does not support *burst* reads or writes, which means that multiple data cannot be retrieved in a single transfer. Consequently, and since read and write accesses are sequential, they can, by accumulating, rapidly catch up the latency penalty inherent to AXI4, which, on the other hand, will take advantage from its parallel data access capabilities when a large memory zone is involved.

Conversely, AXI4 will be less efficient at handling smaller data sets, which is amplified by the fact that it does not benefit from a higher-level interface to manage allocation offsets for individual or fragmented data, which should for now be manually handled by the user, as it has not yet been implemented either in the Syfala API.

Finally, AXI4-Lite adapters are always implemented in the Programmable Logic with Vitis HLS, alongside the DSP IP, meaning that it uses a lot more resources than its AXI4 counterpart, for which the actual memory is in the external DDR.

5.2 Adding External Control Interfaces

One additional reason to provide a custom ARM executable would be to bridge the DSP application with the rest of the system, both in terms of hardware and software components. In a baremetal context, this could mean for instance using the Serial or SPI interfaces to control DSP parameters or, on embedded Linux, taking advantage of USB or Ethernet interfaces to implement MIDI and/or OSC control.

```
[...]

static int gain_handler (
               [...]
            lo_arg** argv
               void* user_data
){
    XDSP* dsp = (XDSP*)user_data;
    float gain = argv[0]->f;
    XDSP_Set_gain(dsp, *(u32*)(&gain));
    return 0;
}

int main() {
    [...]
    auto osc = lo_server_thread_new("8888", error_hdl);
    lo_server_thread_add_method(osc, "/gain",
        "f", gain_handler, &dsp
    );
    lo_server_thread_start(osc);
    [...]
    Status::ok("Application ready, now running");
    while (...) {usleep(5000);}
    return 0;
}
```

The code above shows a basic implementation of OSC control in a custom ARM executable, running in an embedded Linux context, using the liblo library.[11] As one can see, the AXI4-Lite data exchange (which can be considered as 'control-rate' here) occurs in this context directly in the OSC callback function `gain_handler`.

6. RESULTS AND PERFORMANCE

We were able to conduct experimentation on a few complex audio programs, which were initially difficult to implement on the Zybo Z20 development board using FAUST. These examples include Wave Field Synthesis [14] algorithms, capable of supporting up to 10 sources for 256 output channels, biquad-based modal Reverb [15] (up to 18,000 modes), FIR filters (10 filters with 4000 coefficients).

It is difficult to assess the performance improvements between the design yield by FAUST and the design obtained from C++ as it heavily depends on the way the Faust and C++ programs have been optimized. In general, the performance can be improved by a factor of 10 (in terms of resource usage or latency).

This significantly helped us with the prototyping of complex spatial audio systems. For this, we took advantage of the FPGAs reprogramming capabilities and large number of GPIOs to interface them with large speaker arrays. In this context, using C++ allowed us to make significant optimizations, which would not have been possible using FAUST, especially when targeting small FPGA boards, such as the Zybo Z7, which have limited logic resources.

[11] https://liblo.sourceforge.net/

7. CONCLUSION AND FUTURE WORKS

This paper presents a method for writing C++ audio programs to map them efficiently on Xilinx FPGAs using the Syfala toolchain. It can help C++ programmers to take advantage of FPGA capabilities in terms of performances and latency. It also enables the implementation of costly algorithms such as modal reverbs, which are generally too computationally expensive to be run on a CPU in real time.

Finally, being able to use Syfala with C++ will allow us to interface the toolchain with other C++ based frameworks or programming environments (e.g., JUCE, SuperCollider, PureData with Heavy, Max/MSP with gen or Rnbo, etc.).

One of our research direction is to abstract the way pragmas are used in the C++ program so as to include them in the FAUST compiler.

Acknowledgments

This project has been partially funded by the French ANR (Agence Nationale de la Recherche) through the FAST project [12] (ANR-20-CE38-0001) and the Inria/Stanford Plasma Associate Team. [13]

8. REFERENCES

[1] T. Skare and J. Abel, "Gpu-accelerated modal processors and digital waveguides," in *Proceedings of the 2019 Linux Audio Conference (LAC-19)*, Stanford, USA, 2019.

[2] A. HajiRassouliha, A. J. Taberner, M. P. Nash, and P. M. Nielsen, "Suitability of recent hardware accelerators (DSPs, FPGAs, and GPUs) for computer vision and image processing algorithms," *Signal Processing: Image Communication*, vol. 68, pp. 101–119, 2018.

[3] M. Popoff, R. Michon, T. Risset, Y. Orlarey, and S. Letz, "Towards an FPGA-Based Compilation Flow for Ultra-Low Latency Audio Signal Processing," in *Proc. Int. Conf. in Sound and Music Computing, SMC-22*, Saint-Étienne, France, Jun. 2022.

[4] Y. Orlarey, S. Letz, and D. Fober, *New Computational Paradigms for Computer Music*. Paris, France: Delatour, 2009, ch. "Faust: an Efficient Functional Approach to DSP Programming".

[5] M. Popoff, R. Michon, T. Risset, P. Cochard, S. Letz, Y. Orlarey, and F. de Dinechin, "Audio DSP to FPGA Compilation: The Syfala Toolchain Approach," Univ Lyon, INSA Lyon, Inria, CITI, Grame, Emeraude, Tech. Rep. RR-9507, May 2023.

[6] C. Wegener, S. Stang, and M. Neupert, "Fpga-accelerated real-time audio in pure data," in *Proceedings of the International Conference in Sound and Music Computing, SMC-22*, 2022.

[7] T. C. Vannoy, "Enabling rapid prototyping of audio signal processing systems using system-on-chip field programmable gate arrays," Master PhD, 2020.

[8] C. Dragoi, C. Anghel, C. Stanciu, and C. Paleologu, "Efficient FPGA Implementation of Classic Audio Effects," in *Proceedings of the 2021 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*. Pitesti, Romania: IEEE, Jul. 2021.

[9] S. Lahti, P. Sjövall, J. Vanne, and T. D. Hämäläinen, "Are we there yet? a study on the state of high-level synthesis," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 38, no. 5, pp. 898–911, 2018.

[10] R. Nane, V.-M. Sima, C. Pilato, J. Choi, B. Fort, A. Canis, Y. T. Chen, H. Hsiao, S. Brown, F. Ferrandi *et al.*, "A survey and evaluation of fpga high-level synthesis tools," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 35, no. 10, pp. 1591–1604, 2015.

[11] P. P. de Garanderie, "Traitement du signal audionumérique à partir des modulations sigma-delta et pcm," Master's thesis, École Nationale Supérieure Louis-Lumière, 2015.

[12] P. Cochard, M. Popoff, A. Fraboulet, T. Risset, S. Letz, and R. Michon, "A programmable linux-based fpga platform for audio dsp," in *Proceedings of the 2023 Sound and Music Computing Conference (SMC-23)*, 2023, pp. 110–116.

[13] P. Cochard, J. Weber, R. Michon, T. Risset, and S. Letz, "Open Source Ethernet Real-time Audio Transmission to FPGA," INRIA, Tech. Rep. RR-9542, Feb. 2024. [Online]. Available: https://inria.hal.science/hal-04491503

[14] J. Ahrens, *Analytic methods of sound field synthesis*. Springer Science & Business Media, 2012.

[15] J. S. Abel, S. Coffin, and K. Spratt, "A modal architecture for artificial reverberation with application to room acoustics modeling," in *Audio Engineering Society Convention 137*. Audio Engineering Society, 2014.

[12] https://fast.grame.fr
[13] https://team.inria.fr/emeraude/plasma/

ENABLING AFFORDABLE AND SCALABLE AUDIO SPATIALIZATION WITH MULTICHANNEL AUDIO EXPANSION BOARDS FOR FPGA

Maxime POPOFF (maxime.popoff@insa-lyon.fr) [1], **Romain MICHON**[1], and **Tanguy RISSET**[1]

[1]**INSA Lyon**, **Inria**, *CITI*, *UR3720*, 69621 Villeurbanne, France

ABSTRACT

The implementation of audio spatialization and active acoustic control techniques requires a large number of audio channels and significant computational resources, making these techniques costly and hardly accessible. The use of Field-Programmable Gate Arrays (FPGAs) offers a solution by enabling DSP programming within a centralized system with enough computational power and inputs/outputs to handle all audio streams. However, physically interfacing an embedded system with multiple audio channels is a complex process that poses a significant challenge to system accessibility. In this paper, we present two open-source circuit boards facilitating the interfacing between an FPGA and large multichannel systems, both aiming different kinds of approaches. The first board is designed to be cost-effective and easily reproducible. The second involves more complex manufacturing techniques, but it allows us to fully exploit the performances of the FPGA. Each board provides 32 audio channels and has stacking capabilities, allowing for the interfacing of up to 512 channels on a single FPGA. Their integration into the Syfala compilation flow facilitates the implementation of algorithms without hardware concerns. These interfaces aim to enhance accessibility and affordability when implementing audio spatialization and active acoustic control systems.

1. INTRODUCTION

In the fields of active acoustic control and audio spatialization, using a large number of speakers is usually necessary to obtain optimal results.

Some sound spatialization techniques such as Wave Field Synthesis (WFS) [1] can imply the use of thousands of speakers in specialized concert halls [2–4]. The usual approach in handling real-time audio signal processing systems with such requirements typically involves employing a CPU [1] alongside a multichannel audio interface. More recently, accelerators such as GPUs [2] that provide high parallel data processing, have proven to be very efficient with WFS [5] and Active Noise Control (ANC) [6] systems. However, both GPU and CPU implementations rely on external audio interfaces, which implies a significant latency due to buffering. Moreover, the availability of audio interfaces offering more than 32 audio outputs is limited and often comes at a high cost. [3] To allow for more channels, some applications such as Belloch et al. [5] use a combination of multiple audio interfaces, but this approach remains very expensive and depends on the CPU processing capabilities. The same limitations apply when using Ethernet audio interfaces connected in parallel [7]. Other specialized systems employ proprietary multi-channel signal processing systems which are often expensive and not flexible [8].

Field-Programmable Gate Arrays (FPGAs) are very powerful embedded platforms that have been increasingly used in recent years for real-time audio Digital Signal Processing (DSP) applications [9–11]. They provide unique performances in terms of audio latency, computational power, and interfacing.

The use of an FPGA can reduce system expenses by allowing all audio channels to be managed by a single embedded system and by providing enough resources to compute audio samples at a high sampling rate. The high degree of parallelization of FPGAs and their abundance of GPIO [4] pins enables the use of a large number of digital audio channels, making them especially interesting for sound spatialization and acoustic control techniques [12, 13].

However, in order to connect multiple audio inputs/outputs to an embedded system, it is necessary to perform analog-to-digital conversion of the signal with an audio codec, potentially amplify it, and wire each channel. These constraining steps make the practical real-time implementation of massive multichannel algorithms hard to reach. Most multichannel ANC practical implementation on FPGA are either using expensive ADC/DAC [5] [14] or tested on fewer channels [15].

In this paper, we propose to facilitate the use of FPGAs in multichannel systems by developing two different audio expansion boards.

The first one, referred to as the "Frugal Board," facilitates the construction of systems with a high number of outputs at a minimal cost. It focuses on affordability and can be used with a low-cost FPGA-based board due to

[1] *Central Processing Unit*
[2] *Graphics Processing Units*

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[3] See L-ISA: https://l-isa.l-acoustics.com – All URLs in this paper were verified on March 8, 2024.
[4] *General Purpose Input/Output*
[5] *Analog to Digital/Digital to Analog Converters*

its PMOD [6] connectors available on entry-level Digilent hardware.

The second board, named "Ultra Low Latency Board" (or ULL), aims to leverage the full potential of the FPGA by employing an ultra-low-latency audio codec capable of achieving sampling frequencies up to 768kHz and latency as low as $11\mu s$. It is designed to operate with Xilinx's Ultrascale+ FPGAs, which are higher-end compared to those used with the Frugal board.

We then detail how their integration into the Syfala toolchain [16] simplifies their usage by providing a compilation flow from high-level audio DSP specifications down to FPGA bit-streams. Finally, we describe two application that we are currently testing for each board.

Each platform is open-source and available on GitHub. [7]

2. FRUGAL BOARD

In [17], we demonstrated the feasibility of designing a frugal multichannel platform by employing low-cost audio codec. We used an affordable Zybo Z7-20 [8] ADM/Xilinx FPGA development board coupled with 32 MAX98357A [9] codecs to make a WFS system for less than 800 USD. The MAX98357A codecs integrate digital-to-analog converters and built-in 3 watts amplifiers. Communication with these codecs occurs through the I2S [10] TDM [11] protocol, enabling communication with multiple codecs through a single wire and significantly reducing the required number of GPIOs.

However, connecting the 32 codecs necessitates substantial wiring, signal buffering and shielding, and non-trivial power management. In our prior work [17], we built the interface board that embeds the codecs using a dot matrix PCB [12] and hand-soldered cables, a time-consuming and non-robust process that resulted in erratic system operation and hard to debug problems.

To improve reliability, we present a custom-designed PCB in Fig. 1 aiming to enable individuals and institutions with limited resources to build a multichannel platform on FPGA. Two different manufacturing techniques are possible: either using existing commercial Adafruit breakout boards [13] allowing for hand-soldering the audio codecs and a complete DIY fabrication; or using specialized equipment to solder the codec directly to the board to enhance integration.

2.1 DIY Fabrication Approach

A reproducible board implies that it should be manufacturable using basic PCB fabrication techniques, without the need for advanced industrial machinery. We followed

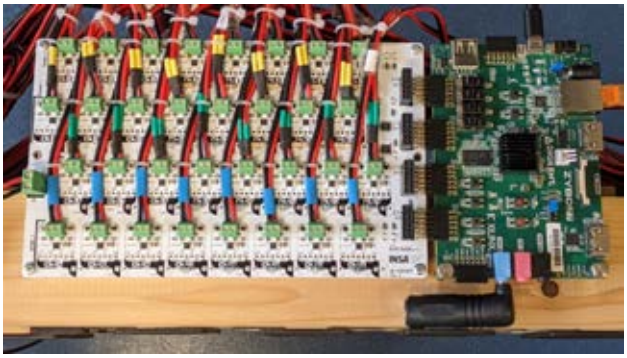


Figure 1. Frugal Board (white) with 32 wired speakers, connected to a Zybo Z7-20.

this requirement by using sufficiently wide PCB track widths to allow for high tolerance during printing, sufficiently large drill holes for manual drilling, routing tracks only on the two outer layers, and soldering all components on one side to facilitate potential hand soldering.

A major challenge of this constraint is that the MAX98357A audio codec has a footprint that is too small for manual soldering and requires the use of a specific equipment. Fortunately, we can use Adafruit breakout boards that integrates the codec as well as a few other essential components soldered onto a small PCB. This PCB can then be hand-soldered onto our board to connect the audio codec to the FPGA (see the top of Fig. 2).

All these constraints make it possible to assemble this board entirely by hand.

2.2 Direct Component Integration Approach

As manufacturing techniques become more accessible, it may be worthwhile to have the board manufactured and the components soldered by a specialized firm. In this case, using Adafruit breakout boards may be more complicated and expensive than purchasing the codec directly. Therefore, in addition to connectors for Adafruit breakout boards, the PCB also includes the footprint of the MAX98357A and peripheral components. This allows for the direct soldering of the audio codec and facilitates integration (see the bottom of Fig. 2).

Figure 2. Two Frugal Boards side by side with the two different integration: direct MAX98357A audio codec integration (bottom) or using Adafruit breakout boards (top).

[6] *Peripheral Module interface*
[7] https://github.com/inria-emeraude/syfala
[8] https://digilent.com/reference/programmable-logic/zybo-z7/start
[9] https://www.analog.com/en/products/max98357a.html
[10] *Inter-IC Sound (IIS or I2S)* is a serial bus interface standard used to interconnect digital audio devices.
[11] *Time-Division Multiplexing*
[12] *Printed Circuit Board*
[13] https://www.adafruit.com/product/3006

2.3 Stacking Capabilities and Clock Signals Integrity

Multiple boards can be stacked to increase the number of channels on a single FPGA. This is facilitated by the presence of stackable PMOD connectors and the possibility to select other GPIOs than the defaults for communication with the codecs (using jumpers). Using the TDM I2S protocol (as explained in Fig. 3), a 32 codecs board need 6 GPIOs: 4 for the data (one for 8 codecs) and 2 for the clocks (shared between all the amplifiers). Bit Clock and Word Select Clock can be shared between stacked boards, so each added boards will only need 4 GPIOs for data (TDM I2S wires in Fig. 3). Knowing that the board can access 32 GPIOs from the FPGA, we can connect up to 7 Frugal Boards to a Zybo Z7-20, allowing for 224 output channels.

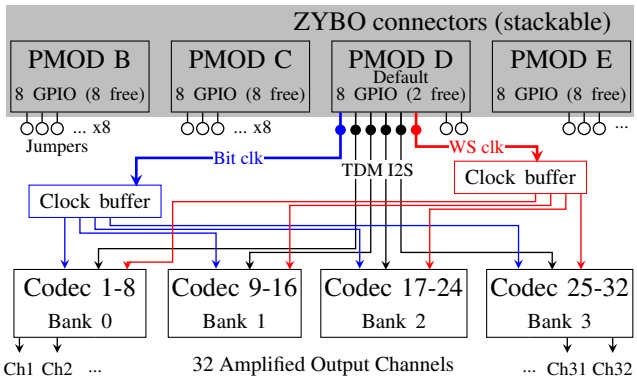


Figure 3. Frugal Board I2S Wiring

To ensure signal integrity and the possibility of connecting multiple boards in parallel to a single FPGA, it is necessary to use a clock buffer on the clock wires. We choose the low cost and low jitter 551S [14] clock buffer that takes the form of a hand-solderable surface-mount component accompanied by damping resistors.

3. ULTRA LOW LATENCY BOARD

The computational power and parallelization capabilities of FPGAs permit the implementation of powerful multichannel algorithms with unmatched low audio latency. While the previous section focused on low cost interfacing, the board described here enables ultra-low latency audio DSP. In our previous work [9], we highlighted the significant role of the audio codec and the associated antialiasing filters, as it is one of the contributors to latency.

To leverage the computational power of the FPGA, it is counterproductive to use inexpensive codecs that introduce excessive latency. Our preliminary work identified the ADAU1787 [15] ultra-low latency codecs that can operate at up to 768kHz and achieve a latency of $11\mu s$ from the analog input to the analog output using the Syfala compiler [9]. Moreover, while the Frugal Board in section 2 only featured codecs capable of synthesizing an audio signal (output channel), the ADAU1787 also hosts audio inputs.

For the experimental tests in [9], we used evaluation boards [16] provided by Analog Devices. Building a 32-input/output audio interface can be done using 16 of these evaluation boards (each codec managing 2 channels). However, this approach poses several problems: (i) a prohibitively high cost (~8000 USD), (ii) a bulky system, due to the form factor of these boards featuring multiple connectors and peripherals unnecessary for our applications, and (iii) substantial wiring, leading to reliability issues.

We propose a custom PCB in Fig. 4 with embedded ADAU codecs that addresses these three issues and facilitates the use of these codecs.

Figure 4. 32 Channels Ultra Low Latency FPGA interface stacked on the Genesys ZU-3EG. The motherboard embeds 16 breakout boards with ADAU1787 codecs.

3.1 Modular Approach

The ADAU1787 codecs are designed for portable devices such as smartphones and digital cameras, thus they are extremely compact. Their small footprint requires advanced equipment for integration, leading to costly PCB printing techniques.

Designing a board with 16 codecs soldered onto it leads to high production costs and several disadvantages, including the inability to individually replace a malfunctioning codec and the impossibility of making later interface changes (e.g., audio port modifications, filter additions, etc.).

For these reasons, we opted for a two-PCB fabrication approach. The first PCB (breakout board) exclusively embeds the codec and its peripheral components (Fig. 5).

[14] https://www.renesas.com/us/en/document/dst/551s-datasheet
[15] https://www.analog.com/en/products/adau1787.html
[16] https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/eval-adau1787z.html

Figure 5. ADAU1787 breakout board.

It employs costly manufacturing techniques but remains minimalist to reduce costs per codec. The second circuit (motherboard) incorporates all the necessary connectors (audio and FPGA) and enables connection to 16 codec PCBs. This setup facilitates the replacement of malfunctioning codecs and eases interface changes in the future, leveraging existing codecs.

3.2 Audio Interface

The ADAU1787 embeds 2 single-ended input channels and 2 output channels, either differential or on an amplified and filtered headphone port. As shown in Fig. 6, on our board, the single-ended input and differential output are connected to DB25 connectors, with 8 channels per connector (4 input connectors and 4 output connectors for a total of 32 channels). These connectors are widely used in the field of audio and facilitate the connection to an amplifier or the use of a multicore cable with jack or XLR connectors.

The headphone output channels are directly routed to a 3.5mm stereo jack port for direct use with headphones.

3.3 FPGA Interface

To communicate with the FPGA, we use an FMC connector.[17] FMC is a standard that defines a compact electromechanical expansion interface for a daughter card to an FPGA baseboard. Most FPGA evaluation boards feature an FMC connector. By using this port, we ensure mechanical and electrical compatibility of our board with a large portion of commercially available FPGA boards.

On the Genesys ZU-3EG,[18] this port allows us to access up to 36 pairs of differential or 72 single-ended signals. Each codec requires 2 dedicated data wires (in and out), and 5 shared wires: 3 clocks (word select, bit clock, and master clock) and 2 wires for the I2C[19] bus. Thus, connecting 16 codecs require 37 wires that can fit on the FMC connector with signals in single-ended mode.

3.4 I2C Configuration

Unlike MAX98357A codecs in section 2, which are very simple and don't have software configuration, the ADAU1787 codecs require configuration via the I2C protocol at each boot. In order to configure each codec independently, they must all have a unique I2C address. The

[17] https://www.samtec.com/standards/vita/fmc/
[18] https://digilent.com/reference/programmable-logic/genesys-zu/start
[19] *Inter-Integrated Circuit (IIC or I2C)* is a serial communication bus used for communication between integrated circuits

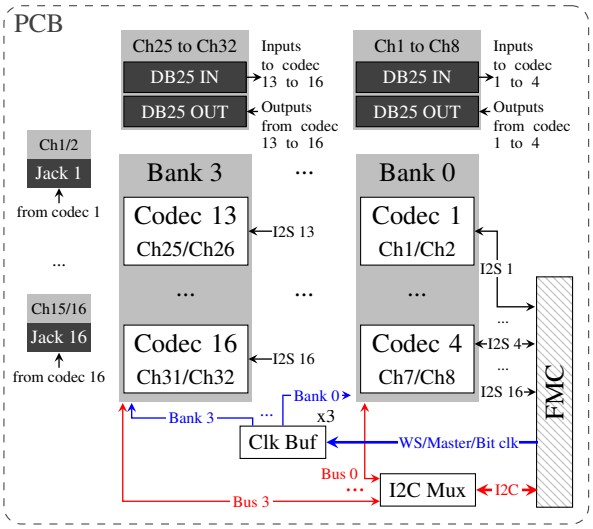


Figure 6. Ultra Low Latency Board I2C and I2S Wiring

codec's address is set by solderable jumpers located on the breakout board. However, the ADAU1787 can only accept four different addresses.

The solution is either to use a new I2C bus for each batch of four codecs, which is costly in terms of GPIO, or to use an I2C multiplexer that can switch on multiple sub-buses from a single I2C bus. If codecs have identical addresses but are on different sub-buses, they can be differentiated. With 16 codecs capable of taking four distinct addresses each, we require at least a 1-to-4 multiplexer. We chose the widely used TCA9548[20] 1-to-8 multiplexer.

3.5 Stacking Capabilities

A single board embeds 16 codecs, providing 32 input/output channels. However, multiple boards can be parallelized on the same FMC port to increase the number of I/Os. Each codec communicates with the FPGA through its dedicated I2S data in and data out wires, resulting in 16 distinct I2S buses on the FMC port. All buses share the same Word Select, Master Clock, and Bit Clock wires.

By employing the I2S TDM protocol as done in Section 2 instead of the traditional I2S protocol, up to 16 codecs can communicate over the same data in and data out wires.

Therefore, it is straightforward to connect codecs in parallel and utilize TDM8 (up to 8 codecs on one bus) or TDM16 (up to 16 codecs on one bus) to communicate with all of them. Consequently, it becomes possible to parallelize up to 16 motherboards, each containing 16 codecs, resulting in a total of 256 codecs or 512 channels.

3.6 Design Considerations and Features

Given that clock signals (WS, Master, and Bit Clock) are shared from one GPIO to all codecs, it is wise to add a clock buffer to these wires to increase their strength and improve signal integrity. The buffer must have very low latency jitter, very low propagation delay and very low output skew (e.g., LMK1C1104[21]) to ensure synchronic-

[20] https://www.ti.com/product/TCA9548A
[21] https://www.ti.com/product/LMK1C1104

ity between all codecs. The PCB has also been designed to potentially add damping resistors in series later to help eliminate reflection.

Particular attention has been paid to the routing of tracks on the PCB. Since the tracks are highly parallel and can exceed 40cm, a spacing greater than 3W (where W is the track width) is maintained in order to minimize crosstalk.

The datasheet of the ADAU1787 specifies that it is not necessary to differentiate the analog ground plane (AGND) from the digital ground plane (DGND). We have chosen to use a single ground plane and not to separate them.

To facilitate debugging, each codec is paired with a status LED controlled by the PCA9956 [22] LED multiplexer.

This board should be compatible with any Ultrascale+ FPGA equipped with an FMC port, we also ensured that it is easily stackable with the Digilent Genesys ZU-3EG development board by using the same dimensions.

4. SYFALA TOOLCHAIN COMPATIBILITY

The use of these two interfaces is facilitated by their integration into the Syfala toolchain [9, 16].

The main challenge of using FPGAs for real-time audio signal processing is that they're notoriously complex to program, mostly because of their inherently low-level architecture involving the use of Hardware Description Languages (HDL) such as Verilog or VHDL. Syfala simplifies the programming of real-time audio DSPs on FPGAs using C++ or the Faust [23] programming language.

The number of audio inputs/outputs is directly derived from the Syfala input program and the toolchain automatically generates the adequate FPGA design, including an I2S transceiver with the appropriate number of channels and configurable sampling rate.

A dedicated option added to the compilation chain allows us to specify the use of one of the two interfaces described in Section 2 and 3. A configuration file manages the addresses of all I2C devices and codecs. The toolchain is compatible with Zynq 7000 (Zybo) and Ultrascale+ (Genesys) FPGAs.

This integration allows for the seamless use of these interfaces from the user's perspective, who only needs to focus on the audio DSP algorithm implemented on the FPGA without needing to manage the hardware.

5. APPLICATIONS

5.1 Frugal WFS System

The affordability of the Frugal Board interface opens the door for massive multichannel systems with several hundreds of speakers at a limited cost.

Building upon our previous work [17], we implemented a simple WFS algorithm with Faust on a Zybo Z7-20 with a 32 speakers array (Fig. 7). The WFS algorithm that we used for our system is fairly basic and standard but we obtained very convincing results. Accessibility comes at the expense of (i) a lower sound quality and (ii) a higher latency compared to expensive codecs. The MAX98357A can work at a maximum sampling frequency of 96kHz and introduce a latency of approximately 500μs at 48kHz. Since each codec is identical and managed by a single FPGA, all speakers are subject to the same latency and thus perfectly synchronized and suitable for audio spatialization applications. The built-in 3W amplifier and associated passive speakers demonstrated sufficient sound quality to achieve effective rendering of the audio spatialization algorithms that we tested.

Figure 7. Prototype WFS system using the Frugal Board

5.2 Active Noise Control

The second interface will target more professional applications for spatial audio, especially those involving active control.

We collaborated with École Centrale de Lyon (France) to test sound field Active Noise Control (ANC) algorithms in their listening room (Fig. 8). This room is equipped with 20 speakers and a 6 microphones spherical array, allowing for the study of large-scale algorithms. We implemented two multichannel Filtered-x LMS algorithms to minimize either multi-point pressure or ambisonic components of the sound field [18]. We have tested these DSPs with only 6 speakers and 4 microphones due to the resource limitations of the FPGA at that time. However, our recent optimization efforts should enable us to conduct the test using all speakers and microphones in the room. The DB25 input/output connectors of our board enabled a seamless integration into their system and the power of the Ultrascale+ FPGA coupled with the low-latency ADAU1787 codecs allowed for the implementation of more efficient ANC algorithms.

Figure 8. Experimental setup inside the listening room.

[22] https://www.nxp.com/docs/en/data-sheet/PCA9956A.pdf

[23] https://faust.grame.fr/

6. CONCLUSIONS

In this paper, we presented two expansion boards for FPGA-based platform that facilitate the interfacing of multiple speakers and microphones with an FPGA. The two different approaches of these boards allow for various applications depending on the needs, either by prioritizing cost and ease of fabrication or performance.

We hope that this work will make massive multi-channel low-latency systems more accessible to the audio and music technology research communities. This should allow for the study and testing of many applications that were out of reach with traditional multichannel audio systems until now, especially in the context of active acoustic control and spatial audio.

Acknowledgments

This project has been partially funded by the French ANR (Agence Nationale de la Recherche) through the FAST project [24] (ANR-20-CE38-0001) and the Inria/Stanford Plasma Associate Team. [25]

7. REFERENCES

[1] A. J. Berkhout, D. de Vries, and P. Vogel, "Acoustic control by wave field synthesis," *The Journal of the Acoustical Society of America*, 1993.

[2] T. Ziemer, *Wave Field Synthesis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018.

[3] M. Noisternig, T. Carpentier, and O. Warusfel, "Dispositif de spatialisation sonore 3D à l'Espace de Projection de l'IRCAM," *Acoustique et Techniques : trimestriel d'information des professionnels de l'acoustique*, 2012.

[4] "Experimental media and performing arts center website," https://empac.rpi.edu.

[5] J. A. Belloch, A. Gonzalez, E. S. Quintana-Ortí, M. Ferrer, and V. Välimäki, "GPU-Based Dynamic Wave Field Synthesis Using Fractional Delay Filters and Room Compensation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2017.

[6] J. Lorente, M. Ferrer, M. de Diego, and A. González, "GPU Implementation of Multichannel Adaptive Algorithms for Local Active Noise Control," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2014.

[7] T. Reussner, C. Sladeczek, M. Rath, S. Brix, K. Preidl, and H. Scheck, "Audio Network-Based Massive Multichannel Loudspeaker System for Flexible Use in Spatial Audio Research," *Journal of the Audio Engineering Society*, 2013.

[8] N. Epain and E. Friot, "Active control of sound inside a sphere via control of the acoustic pressure at the boundary surface," *Journal of Sound and Vibration*, vol. 299, no. 3, pp. 587–604, 2007.

[9] M. Popoff, R. Michon, T. Risset, Y. Orlarey, and S. Letz, "Towards an FPGA-Based Compilation Flow for Ultra-Low Latency Audio Signal Processing," in *Proc. Int. Conf. in Sound and Music Computing, SMC-22*, Saint-Étienne, France, Jun. 2022.

[10] T. C. Vannoy, "Enabling rapid prototyping of audio signal processing systems using system-on-chip field programmable gate arrays," Master PhD, 2020.

[11] M. Verstraelen, J. Kuper, and G. J. Smit, "Declaratively Programmable Ultra Low-Latency Audio Effects Processing on FPGA." in *DAFx*, 2014.

[12] D. Theodoropoulos, G. Kuzmanov, and G. Gaydadjiev, "Multi-Core Platforms for Beamforming and Wave Field Synthesis," *IEEE Transactions on Multimedia*, 2011.

[13] ——, "A Minimalistic Architecture for Reconfigurable WFS-Based Immersive-Audio," in *2010 International Conference on Reconfigurable Computing and FPGAs*, 2010.

[14] D. Shi, W.-S. Gan, J. He, and B. Lam, "Practical Implementation of Multichannel Filtered-x Least Mean Square Algorithm Based on the Multiple-Parallel-Branch With Folding Architecture for Large-Scale Active Noise Control," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2020.

[15] A. Klemd, B. Klauer, J. Timmermann, and D. Sachau, "A Flexible Multi-Channel Feedback FxLMS Architecture for FPGA Platforms," in *2021 31st International Conference on Field-Programmable Logic and Applications (FPL)*, 2021.

[16] M. Popoff, R. Michon, T. Risset, P. Cochard, S. Letz, Y. Orlarey, and F. de Dinechin, "Audio DSP to FPGA Compilation: The Syfala Toolchain Approach," Univ Lyon, INSA Lyon, Inria, CITI, Grame, Emeraude, Tech. Rep. RR-9507, May 2023.

[17] R. Michon, J. Bizien, M. Popoff, and T. Risset, "Making Frugal Spatial Audio Systems Using Field-Programmable Gate Arrays," in *Proceedings of the 2023 New Interfaces for Musical Expression Conference*, Mexico City, Mexico, May 2023.

[18] L. Alexandre, P. Lecomte, and M.-A. Galland, "Experimental Active Noise Control Using Faust On FPGA: Comparison Between A Multi-Point and Spherical Harmonics Method," in *Forum Acusticum 2023 - 10th Convention of the European Acoustics Association*, Torino, Italy, Sep. 2023.

[24] https://fast.grame.fr
[25] https://team.inria.fr/emeraude/plasma/

PERSPECTIVES ON IMMERSION THROUGH LASER DOPPLER VIBROMETRY

Roderick BUCHANAN-DUNLOP (Roderick.Buchanan-Dunlop@ed.ac.uk) (0009-0006-6036-908X)[1],
Michael J. NEWTON (mike.newton@ed.ac.uk) (0000-0002-5292-3267)[1], and
Martin PARKER (martin.parker@ed.ac.uk) (0000-0001-6030-9999)[1]

[1]*Acoustics & Audio Group, Reid School of Music*, **University of Edinburgh**, Edinburgh, UK

ABSTRACT

Recording sounds from novel perspectives is of great interest in sound design. Sound design that provides a particularly immersive experience for the listener is increasingly relevant, with widespread use across music, film, computer games and other media modalities. Laser doppler vibrometry (LDV) is a non-contact technique for the measurement of mechanical vibration. LDV is used in a wide range of research and engineering applications, including biophysics, microelectronics, and the automotive industries. In audio and acoustics research, LDV can be used instead of piezoelectric transducers where high sensitivity, non-contact measurements are important to ensure the accurate capture of vibrations over a wide frequency range. However, while its use for data gathering and analysis is very well established, its potential as a creative tool in immersive sound design has seen very little research. In this paper we show that LDV can provide a unique and creatively valuable sonic perspective on sound making objects. We demonstrate how an LDV system can be optimised for such use, and allow high quality vibration recordings to be made and used in the creation of immersive sound and music experiences. The results build on what is possible with commonly used piezoelectric transducers by harnessing the non-invasive nature of LDV. The ability to measure vibration at a single point with very wide spectral resolution, and to focus on the vibration of both large and tiny objects, reveals a new sonic world with significant potential in the development of immersive sound design.

1. INTRODUCTION

'Immersion' has been used to describe many different ideas, technologies and experiences across a range of disciplines. However, there is an emerging consensus, discussed by Agrawal et al [1] in their literature review of immersion, that it is a phycological concept rather than being dependent on a particular technology or system. Murray [2] provides a helpful insight: '*Immersion is a metaphorical term derived from the physical experience of being submerged in water. We seek the same feeling from a psychologically immersive experience that we do from a plunge in the ocean or swimming pool: the sensation of being surrounded by a completely other reality, as different as water is from air, that takes over all of our attention, our whole perceptual apparatus*'.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

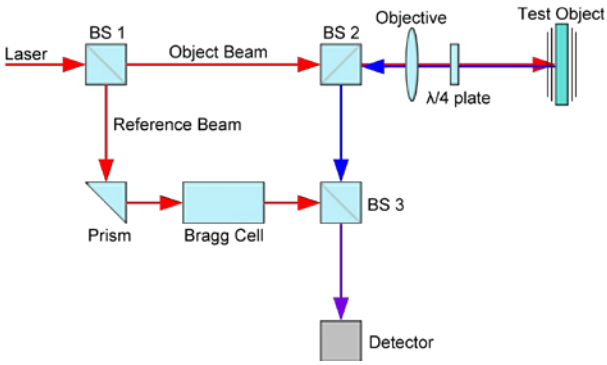


Figure 1. Schematic overview of an LDV system, such as the Polytec MSA-050 unit used in this paper.

When considering the relationship of sound to immersion there are many ways in which immersion can be achieved. This paper is concerned with the use of LDV as a way of creating sounds with an immersive quality to them and does so by addressing two ideas present in Murray's quote. The first is the idea of attention, where unfamiliar sounds have the capacity to facilitate an immersive experience for a listener as they seek to understand what they are listening to. The second is the idea that sound recorded with an LDV invites the listener to immerse themselves in the sound making object, the sonic experience presented is not that of airborne sound but the sound happening at the very surface of the object itself.

2. OVERVIEW OF LASER DOPPLER VIBROMETRY

2.1 Interferometry

A laser doppler vibrometer is a device that uses laser light to measure out-of-plane vibration of physical objects, as illustrated in the schematic diagram in Fig. 1. Laser light, usually from a helium-neon (He-Ne) laser, is first split into two beams via beam splitter BS-1– the object beam, and the reference beam. The object beam is directed at a vibrating object and some of this light is reflected, or backscattered, back into the LDV optics and focussed onto a light-sensitive photo detector. Out-of-plane object mo-

tion (i.e., along the axis of the laser beam) imparts a periodically varying doppler frequency shift in this reflected laser light. The vibrometer optics, particularly beam splitters BS-2 and BS-3, compare this frequency-shifted light with the reference beam to determine the velocity of the test object at the measurement location. Fig. 2 illustrates the relationship between laser light, doppler shifted laser light, and the object velocity.

It is interesting to note that the He-Ne laser emits light of a frequency too high for the photo detector to respond to directly. However, the beat frequency between the object and reference beams, manifested as a light-dark interference pattern, is within the range of the photo detector and this results in an output that is proportional to the speed of the test object. While this arrangement provides a way to measure object speed, it cannot tell whether the object is moving away from the LDV or towards it, as movement in either direction will result in the same intensity at the photo detector. The speed-with-direction information, i.e., velocity, is in fact obtained through additional optics within the LDV, via the Bragg cell. This results in the LDV acting as an interferometer able to determine direction of object movement, also known as a heterodyne interferometer [3].

It is important to note that while velocity is most commonly obtained signal from an LDV system, other signal types can be derived through a variety of methods. For example, with an appropriate optical arrangement, surface displacement may be obtained via phase demodulation of the object and reference beams [3]. Alternatively, digital signal processing in the time or frequency domain can be used, as noted in section 4.8.

2.2 Non-contact surface measurement

The LDV method does not require any physical contact between the sensor and the object being measured, and this is one of its main advantages. Compared with other measurement techniques, such as the use of accelerometers, no additional mass is added to the test object which might influence its vibrational properties. Additionally, hot, lightweight, or moving structures where traditional contact-based instrumentation would change structural dynamics or be difficult to attach can be measured [4]. These advantages, together with excellent velocity resolution, and frequency resolution that can extend into the GHz range on some equipment, have led to its widespread use for vibration measurement. LDV systems are used in research across a diverse range of fields such as automotive engineering [4], biomedical research [5], Micro-Electro-Mechanical Systems (MEMS), and acoustics [6, 7].

This paper uses measurements made with a Polytec MSA-050 LDV system, which is based on a modified Mach-Zehnder interferometer, typical of most LDVs.

3. AUDITORY PERSPECTIVE

3.1 Defining auditory perspective

Given the ability of LDVs to accurately measure vibrations in objects, they are in theory well suited to capturing sound, albeit in a different way to conventional microphones. When a sound making object is excited with energy it vibrates. These vibrations in turn move the air particles around the object causing sound waves to be transmitted through the air. These sound waves can be transduced into electricity by a microphone for recording. While both the LDV and microphone can be used to record the sound produced by an object, they are doing so in fundamentally different ways. The LDV will record the vibration of one specific point on the object's surface. The microphone will transduce the sound in the air around the object, which will be a combination of the sound radiating from all parts of the object (plus any reverberations). While it is possible to position the microphone closer to the object, it will never be able to capture the sound from a single 'point source' on the object's surface. This difference is of fundamental importance as it invites a change in perspective – a crucial aspect of sound recording.

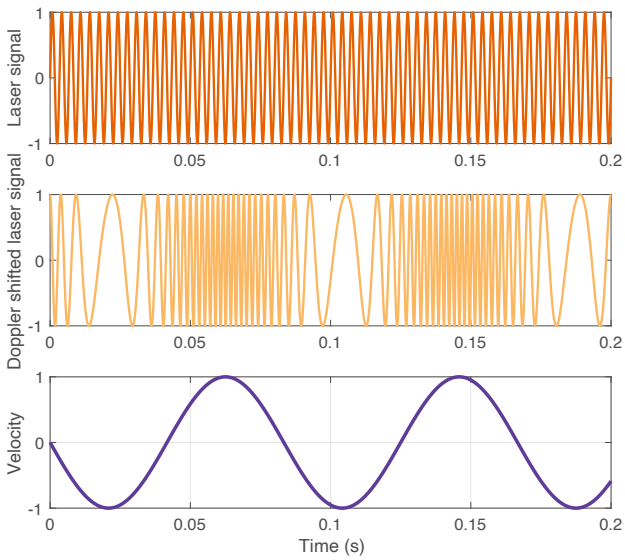


Figure 2. The LDV principle: A point of monochromatic laser light is targeted onto the surface of a vibrating object [top], causing the reflected laser light to be doppler shifted [middle], which is decoded with LDV optics and signal processing to give the out-of-plane velocity of the object at the measurement location [lower] – in this case, a 12Hz sinusoidal oscillation. The vertical axis units are arbitrary in these illustrative examples; similarly, the relative frequencies of object vibration and laser light are illustrative only, being much lower than in a real LDV system.

Auditory perspective is the implied position of the listener relative to the sound source in a recording. Sound designer Karen Collins describes it as follows: '*Auditory perspective is constructed by a variety of techniques that create or reinforce the physical sense of space for the listener through the use of spatialized sound. These techniques combine physical acoustics with psychoacoustics*' [8].

3.2 Unusual auditory perspectives

Loosely defined, 'reverberation' is most often associated with perspective. However, it is instructive to consider

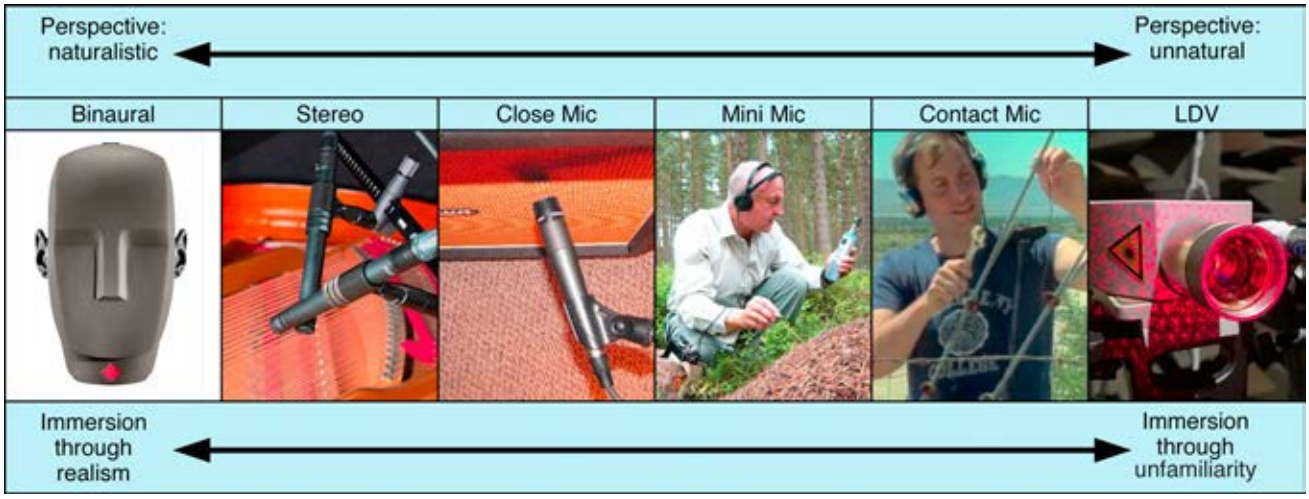


Figure 3. A notional organisation of perspective and immersion for sound design as a function of the audio acquisition tool, which illustrates at once the alignment between perspective and immersion, as well as apparent contradiction.

the contributory factors to reverb, such as echo timing and strength, frequency dependent damping, and channel width, occlusion, obstruction, and exclusion. Together these are interpreted by the listener to give an understanding of their spatial relationship with the sounds they hear. There are many recording and processing techniques, such as binaural and stereo, that aim to impart a realistic, albeit creatively manipulated, sense of perspective in order to define the relationship between the listener and the sound.

However, some audio practices have evolved that allow perspectives that are not recognisably human experiences. For example, sound recordist Chris Watson uses miniature microphones that can be positioned in very close proximity to the sound source of interest: '*When you get microphones in close, into places where you wouldn't want to or be able to put your ears, then the world is revealed in a very different way*' [9]. Watson has put this technique to creative use recording ant hills and glaciers from perspectives that are far beyond our experience as human listeners.

Contact microphones (usually constructed from piezoelectric accelerometers) are another recording method used to capture sound from an unusual perspective, this time from the surface of an object. This makes them relevant to discussion around sound capture using an LDV. The use of contact microphones has found its way into many sound practices including music, sound design, and sonic art. Composers seeking new sounds and approaches to performance embraced contact microphones from the 1950s onwards. John Cage's 'Cartridge Music' (1960) and David Tudor's Rainforest (1973) are early examples that use the distinct perspective afforded using contact transducers to shape and present sound in new ways. Pioneering sound designer Ann Kreober recognised the potential of contact microphones as a way of capturing unusual perspectives on often ordinary sounds: '*I attached the contact microphone to all sorts of things. It sounded as if you were actually inside the thing. As if you were inside a machine. When I attached the Frap* [contact mic] *to a ventilator, a whole new world opened up*' [10].

Kreober made extensive use of contact microphones for many projects including the sound design for the David Lynch film Dune (1984). Contact microphones have also been used by sound artists, such as Jez Riley French, who describes their creative use [11]: '*They reveal, with careful placement... a myriad of different acoustic textures, from the deep, subtle drone of metal structures to intricate movements of plants and insects. I'm fascinated by these hidden elements and the fact that every surface, every object is acting as a filter for the sounding world*'.

3.3 The LDV perspective

While contact microphones offer a unique perspective on sound which has a great deal of creative utility they do have some disadvantages. They add mass to the object they are attached to, and are typically flat and fairly large (circa 20mm diameter), making them awkward to attach to some objects. They are also limited in their frequency response, typically being focussed on the lower half of the human hearing range (up to about 10kHz) at the expense of the extremes of high and low frequencies. Noise can be an issue that limits the range of sources that can be captured. The use of the LDV offers a way to build on the perspective and recording abilities of the contact microphone, but adding value through being non-contact and wider bandwidth. This novel perspective may provide a rich source of audio for immersive sound design, as illustrated in Fig. 3.

4. PRACTICAL CONSIDERATIONS FOR THE USE OF LDV IN SOUND RECORDING

4.1 Introduction

Despite the many advantages of LDV systems in vibration and acoustics measurements, in practice a range of challenges are encountered. These challenges can be especially relevant in audio applications, such as in using LDV signals for immersive sound design. In this section, theoretical challenges around signal integrity and sources of

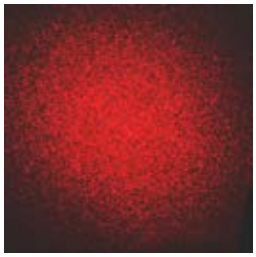


Figure 4. Speckle pattern produced by laser light focussed onto an on object surface [13].

noise are outlined, together with a summary of practical considerations that have been found to reduce or eliminate these issues. Although the LDV device used was a Polytec MSA-050, the challenges and mitigations presented here are universal to the basic method, and will likely apply to many models and implementations of LDV.

4.2 Speckle noise

In any audio recording context noise is present and where it is audible it needs to be managed to ensure usable sound recordings. For LDV applications the surface texture of the object being measured is closely linked to the signal to noise ratio achieved [12]. A major source of noise in LDV systems is that of speckle noise.

Speckle noise is not introduced from an external source but from the use of laser light itself. The coherent nature of laser light makes it an ideal light source for LDVs. However, when this light is reflected by an optically rough surface, where the surface roughness is approximately equal to or greater than the wavelength of the laser light (633nm for the system used here [14]), the result is a distinctive speckle pattern as shown in Fig. 4. The uneven surface scatters the light into wavelets with different directions. Where wavelets converge on the same point the path difference will determine the phase relationship of the detected light. Those in-phase will result in constructive interference and a light speckle, while those out of phase will result in destructive interference and a dark speckle. When the object vibrates the speckle pattern on the photo detector is constantly changing and this becomes a source of broadband noise in the decoded velocity signal.

4.3 Signal drop-outs

A more extreme form of noise exists under conditions where insufficient light is backscattered into the LDV optics, and the velocity decoder cannot determine the object velocity, usually for just a few samples worth of time. Instead, a drop-out occurs which is a brief loss of signal characterised by an audible click and distortion across the frequency spectrum; Fig. 5 provides an example from measurement of a vibrating cymbal head. Several factors may lead to a full drop-out occurring – see section 4.4.

In some measurement applications this behaviour is not overly problematic, but in applications involving audio playback for sound design purposes, it can be a significant issue. Practical setups should aim to minimise this behaviour as far as possible through ensuring strong signal integrity from the LDV unit.

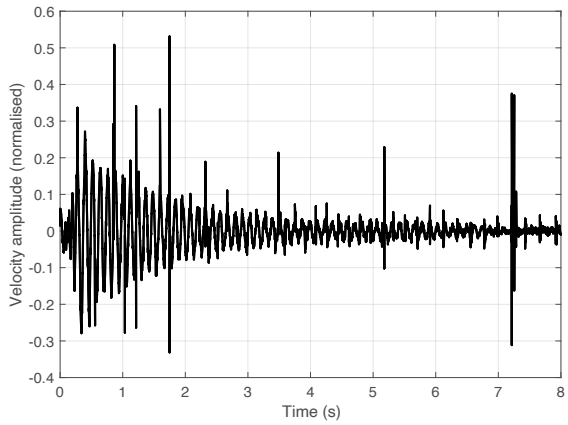


Figure 5. Signal drop-outs seen in an LDV measurement of a vibrating cymbal head. The random signal jumps are the drop-outs, caused by a temporary loss of backscattered light.

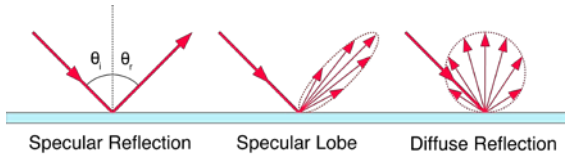


Figure 6. Three kinds of surface reflection that can happen when an LDV laser is focussed on a test object.

4.4 Surface reflection

While both speckle noise and drop-outs are distinctly different kinds of noise that occur for different reasons they share a common cause: the reflective properties of the sound making objects surface

Fig. 6 shows three different kinds of reflection, specular, specular lobe and diffuse. Highly reflective surfaces that produce a specular reflection are not ideal for LDV applications as the beam can easily be directed away from the LDV by the motion of the test object resulting in no signal at the detector. However, rough surfaces that produce diffuse reflections spread the beam over a wide area and result in not enough of the reflected light making its way to the LDV for a reliable signal with a high chance of drop-outs. Furthermore, any light that does make its way back into the LDV is likely to have phase discrepancies leading to speckle noise. A midpoint between these is ideal with a specular lobe reflecting most of the light into the LDV with some margin for additional movement helping to preserve the integrity of the signal. In practice this can be achieved by the use of retroreflective tape or the application of magnesium oxide powder to optically poor surfaces.

Fig. 7 shows a cymbal with a retroreflective tape applied to the surface to reduce drop-outs during recording.

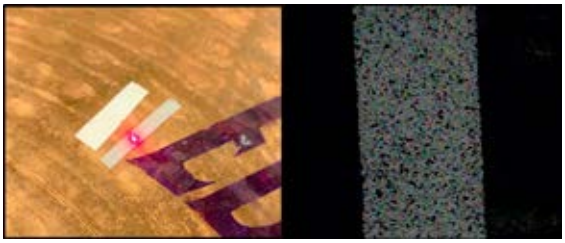


Figure 7. Retroreflective tape [left] applied to the surface of a cymbal, to reduce diffuse reflections (section 4.4) and speckle noise (section 4.2). A 1mm sized portion of the tape is shown through the on-axis LDV camera [right].

4.5 Pseudo-vibration

Pseudo-vibration is another source of measurement noise in LDV systems. Pseudo-vibration takes the form of contributions to the decoded velocity signal due to the laser beam 'scanning' across the optically rough surface of the test object as it vibrates in directions other than perfectly on-axis with the beam [15]. This causes a kind of periodic speckle noise (as per section 4.2) that can manifest as an otherwise plausible contribution to the velocity signal.

Reducing pseudo-vibration noise can be a challenge, depending on the nature of the sound making object. However, the following considerations have been found to be helpful when setting up a recording session for the purposes of capturing object vibration for immersive sound design:

1. When exciting sound from an object, consider the direction that the force or movement is applied and experiment with this. Some directions will usually lead to a similar sonic outcome but with reduced noise form pseudo-vibration.

2. The angle of incidence of the laser beam should be perpendicular, however, experience has shown that often a small repositioning can be enough to reduce noise. During recordings it has been advantageous to tripod mount the LDV sensor head to allow for flexible and stable positioning.

3. Clamping the test object to reduce movement other than the out of plane vibrations of interest helps to reduce the effect of pseudo-vibration noise. However, care should be taken not to dampen the sound of the object any more than is necessary.

4.6 Stand-off distance

Lasers tend to be thought of as sources of pure monochromatic light, but in practice this is not always true. In many cases lasers emit light of multiple frequencies, also referred to as laser modes. The cavity length of the laser employed by the MSA-050 LDV system in this paper is relatively short, and so in theory only one or a maximum of two laser modes are possible [14]. However, manufacturing tolerances, helium-to-neon ratio, pressure, and temperature all contribute to variation of the relative intensities of the two laser modes over time [16].

When setting up the LDV system for audio recording it is therefore best practice to choose a stand-off distance, d_s, that avoids any potential interference between the two laser modes; d_s is the distance from the head of the laser unit to the test object. The safest distances d_s to use are multiples of the laser's cavity length – 204mm in the case of the MSA-050. This distance is also known as the laser coherence repeat length, '*where the light frequencies emitted by the laser are all in phase*' [17]. For the MSA-050 unit, the smallest stand-off distance is 91mm - only possible with a microscope objective. The distances are then calculated by adding integer multiple of 204mm to this, giving optimal stand-off distances of 91mm, 295mm, 499mm and so on.

4.7 Focus

When setting up the LDV the laser should be accurately focused on the object in order to maximise the reflected light. This helps to ensure a good doppler signal at the detector and reliable results. However, there is a downside to sharply focusing the LDV, and that is that the speckle pattern of the reflected light will be at its most clearly defined and has the potential to produce more speckle noise (see section 4.2) as a result [17]. Consequently, in most practical audio recording situations there is a balance to be struck between a good signal level through accurate focus, and slightly defocusing the laser to produce an averaging effect on the speckle pattern which has the result of reducing speckle noise, and producing a cleaner velocity signal.

A related issue is that of optical depth of field, which is the distance range either side of the strict focus plane that is acceptably sharp and able to produce a sufficiently clear interference pattern within the LDV optics. The depth of field D of the LDV optics, in mm, is related to the stand-off distance d_s as follows [14]:

$$D = \pm \frac{d_s^2}{25000} \tag{1}$$

Note that this applies only when the microscope lenses are not being used. Increasing the stand-off distance can therefore be used to increase the depth of field and make measurements of objects with greater displacement ranges. Experience has shown that insufficient depth of field can lead to signal drop-outs (see section 4.3) and that moving the LDV head away from the test object to the next suitable standoff distance can improve signal integrity.

4.8 Types of output signal

4.8.1 Overview

As noted in section 2.1, the native signal type of LDV systems is usually surface velocity. Alternative signals, including displacement and acceleration, can be important in some engineering applications, such as setups also involving surface mount accelerometers; they can also be useful for audio applications. In this section, a brief summary of the issues involved in obtaining and using these alternative signals is presented, as well as brief reflections on

relevance for capturing audio signals for immersive sound design.

4.8.2 Obtaining displacement and acceleration signals from the velocity

Some LDV devices, including the MSA-050 system used for this paper, can output signal types other than velocity, such as surface displacement, or surface acceleration. In a real-time context these signals can be obtained directly from an appropriate optical arrangement involving phase demodulation of object and reference beams [3], or via signal processing of the velocity signal. For the latter case, or where direct displacement decoding is not available, real-time applications tend to employ time-domain differentiation or integration of the velocity signal, usually in the discrete time domain. In offline applications, discrete frequency domain methods provide a convenient alternative approach.

Whichever method is used to obtain displacement and/or acceleration signals from an LDV setup, some general principles are of note in practical applications. Consider the mono-frequency vibration of a mechanical object, of velocity $v(t)$, amplitude A, and frequency f such that

$$v(t) = \Re(Ae^{i2\pi ft}) \tag{2}$$

where $\Re$ denotes taking the real part.

The displacement $x(t)$ is found via time integration as

$$x(t) = \Re\left(\frac{A}{i2\pi f}e^{i2\pi ft} + C\right) \tag{3}$$

for an arbitrary integration constant C, which can usually be ignored. Relative to the velocity, the displacement signal has an amplitude scaled by $1/2\pi f$. In contrast, the acceleration $a(t)$ is found, via time differentiation, to be scaled by a factor of $2\pi f$ (relative to velocity). Across all frequencies, this is equivalent to a 6dB/octave rolloff for displacement relative to velocity, and 6dB/octave gain for acceleration relative to velocity (see Fig. 8). In general, this means that an acceleration signal will be 'brighter' than a velocity signal, and a displacement signal will be 'duller', all else equal.

Where a velocity signal has been sampled digitally, frequency domain discrete integration/differentiation can be employed to obtain the displacement and acceleration signals [18]. Coding examples are available via section 6.1.

4.8.3 Signal types for use in immersive sound design applications

The previous sections 4.8.1–4.8.2 lead to a natural question when using an LDV the system to record sound – which signal should be used? As might be expected, the answer depends on the sound design goals of the given application.

In more conventional sound recording different microphone transducers have been developed that help to provide some context for these signals. Condenser microphones output a signal proportional to the displacement of the diaphragm, whereas ribbon and dynamic microphones output a signal proportional to the velocity of the

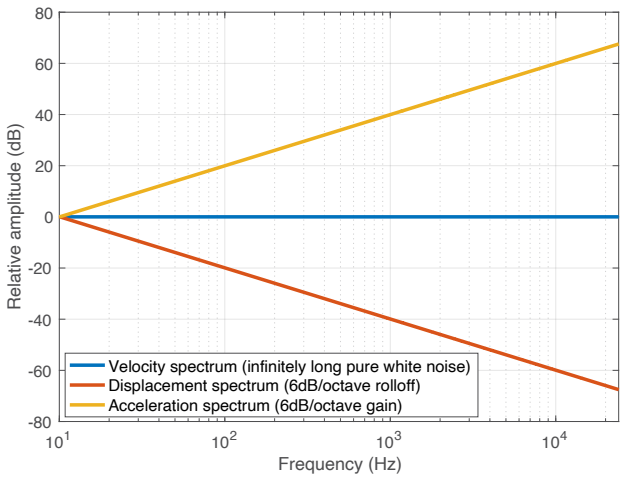


Figure 8. The relative spectral differences between velocity, displacement, and acceleration signals, for a hypothetical white noise velocity signal of infinite length.

diaphragm. In practice there is a complex interplay between the transducer type and the stiffness, damping and mass control of the diaphragm [19], as well as the accompanying electronics. While the ideal microphone would seem to require a flat frequency response this is technically hard to achieve and often creatively undesirable. Microphones will typically have a 'sound' that is imparted onto the recorded material primarily due to an uneven frequency response, whether the result of shortcomings or by design.

As with microphones where context and subjective goals will guide choice, our LDV recordings have shown that that both displacement or velocity may be chosen based on context. Our current practice is to capture both, although upon listening, there is usually a clear preference for the given application.

The difference between the displacement and velocity signals can be heard clearly in example recordings of a cymbal being gently struck – see section 6.1. The displacement signal has significant low frequency content from the surface vibration of the cymbal, whereas the velocity signal emphasises higher frequencies.

5. LDV WORKFLOW FOR AUDIO AND IMMERSIVE SOUND DESIGN APPLICATIONS

5.1 Data format

Using a piece of equipment designed for scientific research presents several workflow challenges when attempting to use it for immersive sound design applications. The first is the ASCII file format used by the LDV acquisition software for data export.

The typical file format of LDV data can be seen in Fig. 9, which shows the MATLAB data import window for a displacement signal (column 2). To turn this into a format that can be used in sound design applications, the first step is to convert the data into a 16-bit signed audio file at the same sample rate the data was captured at, building in 6dB of headroom. In many cases it is useful to convert

	SourceFileName	Scan
	Number	Number
1	Source File Name:	Scan
2	Signal:	Time – Vib Displacement – Samples
3		
4	Time	Time Signal
5	[s]	[m]
6	0	-6.495247e-8
7	1.9531250437e-5	-5.389208e-8
8	3.9062500873e-5	-4.462201e-8
9	5.859375131e-5	-7.45105e-9
10	7.8125001746e-5	-6.162111e-8
11	9.7656252183e-5	-9.959821e-8
12	0.00011718750262	-2.084298e-9
13	0.00013671875306	-3.187229e-8
14	0.00015625000349	-1.2558e-7
15	0.00017578125393	1.600358e-8
16	0.00019531250437	-9.546134e-8
17	0.0002148437548	-9.8825e-9

Figure 9. File format of raw ASCII exported displacement data from the Polytec LDV software.

Figure 10. A practical example of the LDV in use for sound design applications: capturing the subtle mechanical sounds of a vibrating bike frame.

from the 51.2kHz sample rate of the Polytec platform to a more conventional 48kHz rate. Supplementary materials are provided via section 6 to demonstrate this via example MATLAB coding scripts.

5.2 Monitoring of signals during acquisition

A key issue when using the LDV to make audio recordings for use in sound design applications is that of monitoring. In conventional audio recording the signal of interest is carefully monitored in real time at the end of the signal chain, allowing any issues with the signal, either technical or aesthetic, to be addressed before a recording is made. However, monitoring on a system like the MSA-050 is limited to a crude 'level' reading of the Doppler signal strength, rather than giving the ability to instantly audition the signal from a qualitative standpoint. The only way to monitor the LDV output is to capture the data and then go through the conversion process established above and listen to the results.

To improve on the time-consuming process outlined above, a high-quality audio interface was integrated into the system. Instead connecting the LDV's decoded velocity (or displacement) output to the standard junction box, it is connected to the analogue line input of an RME Fireface UCX, which can accommodate levels up to +24dBu – equivalent to 12V, compared with the 10V maximum of the junction box. This offers a way to simplify the process as the signal can be digitised directly at a conventional audio sample rate, negating the need for sample rate conversion and data manipulation. Additionally, metering is accurate, making it easier and quicker to set the velocity or displacement range relative to the maximum input of the interface. The signal can be monitored on headphones, providing instant feedback on issues encountered when setting up. Speckle noise, drop-outs, and pseudo-vibration are all easily diagnosed and can then be addressed. Being able to hear the signal in real time is particularly helpful when focusing the laser and attempting to find the best balance between maximising signal level through sharp focusing of the laser, and defocusing to reduce speckle noise. In fact, this is such a quick and intuitive method that monitoring in this way may prove to be a beneficial addition to LDV setups even when audio recording is not the primary aim, given the sensitivity of human listeners in auditioning audio signals.

While this provides instant feedback and negates the need for conversion, the data export method is still relevant to future use cases where regions of the frequency spectrum above the capabilities of audio range converters are of interest. Automated gathering of measurements for multiple points is also more easily achieved using the full Polytec software package.

6. USE OF OF LDV-BASED AUDIO RECORDINGS

6.1 Audio quality and perspective

Using the methodology outlined above recordings of everyday sound making objects can be made. While care is needed to capture sounds with the system, tests have shown that in ideal situations, a signal to noise ratio of around 60dB is achievable. This makes recording for creative audio applications a realistic proposition.

The unique perspective offered by LDV has some similarities to contact microphone recording – the lack of any reverberation gives resulting sounds a remarkably close and immediate quality. However, the spectral resolution is much wider than can be achieved with a contact microphone and, subjectively, the recordings sound more detailed as a result. The non-contact nature of the technique makes it possible to record a wide variety of objects and materials. An illustrative example of a real-world sound design application is provided in Fig. 10.

Sound examples and coding examples can be found via:

- https://github.com/self-noise/SMC2024-LDV

6.2 Cost considerations

The cost of a typical LDV system is high when compared to more established recording tools, such as contact microphones, mics, and audio interfaces – the LDV system used in this work cost over €130k when new. This presents a barrier to widespread use for sound recording, at least when considering established delivery platforms.

However, technological advances mean that this cost will come down over time, and it is possible that a streamlined

system, optimised for creative audio applications, could be designed at a significantly reduced cost. For example, as has been shown in section 5.2, conventional audio interfaces can be used to replace a significant portion of the signal acquisition system.

For the time being, the benefits of the technology and the perspective offered can be made available through the creation of sample libraries. Specific objects, themes, and recording approaches may be chosen to make the most of what this uniquely immersive sound recording approach has to offer.

7. CONCLUSIONS

In this paper we have shown that recording sound with an LDV system presents a range of technical challenges. In particular, the primary sources of noise – speckle noise, drop-outs, and pseudo-vibration – are fundamentally different to noise encountered in conventional sound recording. However, with an understanding of the operating principles and practical considerations, a working method has been established that allows high quality sound recordings to be made. In investigating the challenges involved and developing mitigations and working practices to overcome them, LDV can be used creative sound applications.

With an LDV, sonic artists can capture the character of vibration at specific points on the surface of a wide range of materials. An LDV brings perspectives that are not possible with contact or air-based microphones. As Kroeber pointed out with contact microphones, a whole new sonic world has opened up [10]. The sounds we have already captured reveal the immersive potential of this technique, bringing the ear closer to the sonic behaviour of materials that may be inaudible or impossible to capture otherwise. We see exciting potential to capture sound that may immerse listeners within unfamiliar contexts.

8. REFERENCES

[1] S. Agrawal, A. Simon, S. Bech, K. Bæntsen, and S. Forchhammer, "Defining Immersion: Literature Review and Implications for Research on Audiovisual Experiences," *J. Audio Eng. Soc*, vol. 68, no. 6, pp. 404–417, 2020.

[2] J. H. Murray, *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*, updated edition ed. The MIT Press, 2017.

[3] P. Chiariotti, C. Rembe, P. Castellini, and M. Allen, *Laser Doppler Vibrometry Measurements in Structural Dynamics*, R. Allemang and P. Avitabile, Eds. Springer, 2020.

[4] S. Rothberg, M. Allen, P. Castellini, D. D. Maio, J. Dirckx, D. Ewins, B. Halkon, P. Muyshondt, N. Paone, T. Ryan, H. Steger, E. Tomasini, S. Vanlanduit, and J. Vignola, "An international review of laser Doppler vibrometry: Making light work of vibration measurement," *Optics and Lasers in Engineering*, vol. 99, pp. 11–22, 2017, laser Doppler vibrometry.

[5] K. Kroschel, Ed., *Laser Doppler Vibrometry for Non-Contact Diagnostics*. Springer, 2020.

[6] A. Zhang and J. Woodhouse, "Reliability of the input admittance of bowed-string instruments measured by the hammer method," *The Journal of the Acoustical Society of America*, vol. 136, no. 6, pp. 3371 – 3381, 12 2014.

[7] I. Wollman, C. Fritz, and J. Frelat, "On the Characterization of Vibrotactile Feedback in Violinists' Left Hand: A Case Study," *Acta Acustica united with Acustica*, vol. 101, no. 2, pp. 360–368, 2015.

[8] K. Collins, "Sonic Subjectivity and Auditory Perspective in Ratatouille," *Animation*, vol. 8, no. 3, pp. 283–299, 2013.

[9] M. Blackwell, "Tone Glow 057: Chris Watson," 3 2021. [Online]. Available: https://toneglow.substack.com/p/057-chris-watson

[10] G. Helbling, "Filmsound Articles: Ann Kroeber and Alan Splet," 4 2020. [Online]. Available: https://www.guidohelbling.com/filmsound-articles-ann-kroeber-alan-splet-and-david-lynch

[11] L. Pisano, "The epiphany of sound (through) objects: Jez Riley French," *Blow Up Magazine*, vol. 177, 2013. [Online]. Available: https://edin.ac/3IC2Ir2

[12] P. F. O'Malley, J. F. Vignola, and J. A. Judge, "Amplitude and Frequency Dependence of the Signal-to-Noise Ratio in LDV Measurements," vol. Volume 1: 23rd Biennial Conference on Mechanical Vibration and Noise, Parts A and B, pp. 545–549, 2011.

[13] Wikipedia, "Speckle (interference)," 5 2011. [Online]. Available: https://en.wikipedia.org/wiki/Speckle_(interference)

[14] Polytec-GmbH, "User Manual for Compact Laser Vibrometer (NLV-2500-5)," 2014.

[15] P. Martin and S. J. Rothberg, "Methods for the quantification of pseudo-vibration sensitivities in laser vibrometry," *Measurement Science and Technology*, vol. 22, no. 3, p. 035302, 2011.

[16] "Application Note: Center Wavelength, Spectral Linewidth, and Coherence Length of Helium-Neon (HeNe) Lasers," 2024. [Online]. Available: https://pacificlasertec.com/technical-info

[17] K. B. Gatzwiller, K. B. Ginn, A. Betts, and S. Morel, "Practical Aspects of Successful Laser Doppler Vibrometry based Measurements," in *Proceedings of the German Acoustical Society Conference (DEGA)*, 2003.

[18] A. Brandt and R. Brincker, "Integrating time signals in frequency domain – Comparison with time domain integration," *Measurement*, vol. 58, pp. 511–519, 2014.

[19] G. S. K. Wong, "Springer Handbook of Acoustics: Chapter 24 (Microphones and Their Calibration)," T. D. Rossing, Ed., 2014, pp. 1061–1091.

CONDITIONING METHODS FOR NEURAL AUDIO EFFECTS

Riccardo SIMIONATO (riccardo.simionato@imv.uio.no)[1] and
Stefano FASCIANI (stefano.fasciani@imv.uio.no)[1]

[1]*Department of Musicology*, **University of Oslo**, Norway

ABSTRACT

Machine learning techniques have become a common approach for modeling analog audio effects. Black-box and hybrid solutions have been applied to a large variety of audio effects. Audio effects usually incorporate user-controllable parameters, and how to infuse this information into the networks is still a challenge. Feature-wise Linear Modulation is a popular conditioning method, but its use in audio effect modeling is still limited. This technique involves an affine transformation via learnable coefficients based on conditioning information. This study compares this approach with other proposals used in this field, such as gated activation. In addition, the control parameters may present a nonlinear relationship with the effect's associated sonic response. Therefore, the investigation also considers nonlinear mapping. This case study investigates two types of analog audio effects: distortion and dynamic range compression. Results indicate the conditioning layer leads to better performance if placed at the end of the architecture, and the Feature-wise Linear Modulation method outperforms other approaches. In addition, nonlinear mapping can be beneficial for cases with strong nonlinear relationships between parameters, such as the overdrive effect.

1. INTRODUCTION

Machine learning has become widely used in audio modeling effects. In an early attempt, a multilayer feedforward network was utilized to learn a high-pass filter and the Ibanez Tube-Screamer distortion effect [1]. Subsequently, convolutional-based networks have been applied for a vacuum-tube amplifier [2] and distortion pedals [3]. Here, conditioning information was infused using a gated activation (GA) as used in the original Wavenet network [4]. Recurrent networks have also been applied for modeling distortion pedals and amplifiers [5–7]. In the case of [6, 7], the conditioning information was incorporated as an extra input to the network. The mentioned works considered one parameter control. A larger number of parameters are considered in [8, 9], where a sequence-to-sequence-based model, based on recurrent networks, exploited the sharing of internal states for the case of an optical compressor. One network encodes the conditioning information, and in this case, some past samples of the signal in its internal states are also shared with the decoder when inferring the output. Two parameters were initially used [8] and later extended to four [9], particularly including parameters determining a time-variant response, such as attack and release time. Still, in the case of another optical compressor, a Temporal Convolution network (TCN) has been used [10] together with the Feature-wise Linear Modulation (FiLM) method [11]. In this case, two parameters were included, but they did not have time characteristics. The FiLM method computed two vectors from the conditioning information to perform linear mapping of the input or some layer's output. FiLM layers learn how to manipulate the control parameters to create a map and decide how this information should influence the network. An activation function such as ReLU or sigmoid can be placed at the end to decide how much information should pass to the inferring network.

Machine learning modeling of nearly all types of analog effects has been investigated. However, not all the proposed models integrate the variable control parameters typically found on audio effect units. A comprehensive model of an audio effect should include user-controllable parameters that determine variations in the sound alteration process. These parameters should allow for continuous changes within their range, determining an immediate change in the model's response. To address this gap in machine learning modeling of audio effects, we further investigate approaches for conditioning with control parameters artificial neural network models of different effects. We compare three approaches used in the field, such as the FiLM and GA layers, against concatenating the conditioning information with the input samples representing the baseline. FiLM and GA are placed at different stages in the architecture to compare when the conditioning information is more beneficial for prediction accuracy. Finally, control parameters may present nonlinear relationships, resulting in a variation of the sound alteration process of the audio effect. Therefore, including nonlinearities in the conditioning block can be beneficial. We use two audio effects to evaluate the various conditioning approaches: overdrive and compressor. For each effect, we select only two variable parameters to condition the networks: the level and tone for the overdrive and the ratio and threshold for the compressor.

Section 2 presents our experiments' methodology, architectures, and datasets. Section 3 detailed the obtained results and the consequent discussion, while Section 4 concludes the paper with a summary of the findings.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. METHODS

The experiments reported in this paper aim to explore different conditioning methods for artificial neural network modeling of audio effects. The architecture is based on state-space models (SSMs) [12]. In particular, we employ the diagonal state space model variant, the S4D layer [13].

We explore the placement of conditioning layers at different stages of the neural network. Specifically, we consider three configurations labeled *pre*, *post*, and *pre-post*. The first configuration presents the conditioning layer before the S4D layer, the second after the S4D layer, and the last one includes a conditioning layer before and after the S4D layer. At the same time, we evaluate different network architectures for the conditioning layer.

In particular, we use the FiLM and the GA layers. The first consists of computing two vectors from the conditioning information using a linear function to perform a linear mapping of input or some layer's output. The second involves learnable convolution weights for the conditioning vector and the latent space computed from the input vector. The sigmoid is applied to the outputs of the first convolution, while the hyperbolic tangent is applied to the second one. Finally, the two results are element-wise multiplied together. The FiLM layer originally used FC layers to predict the coefficient for the transformation. We expand the methods using convolution as in the GA. These approaches are compared with a baseline we obtain by training similar S4D-based architecture without conditioning layers and with conditioning values provided as input and the array of audio samples. Furthermore, we carry out experiments exploring different transformations inside the FiLM layer. Linear mapping preserves the dimension of any affine subspaces. Still, in the context of audio effect modeling, control parameters may present nonlinear relationships with the sound-altering process of the audio effect. For this reason, we explore nonlinear transformations to assert if increasing the transformation complexity benefits the conditioning process. In particular, we selected transformations of odd orders due to their anti-symmetric characteristic.

As mentioned, the architecture selected for our experiments is based on SSMs, particularly the S4D model. These operate similarly to recurrent neural networks but showed remarkable results on sequence modeling tasks and improved accuracy also in audio effect modeling [14]. The remarkable ability of SSMs to capture long-range dependencies stems from their utilization of a specific state matrix known as the HiPPO matrix [15]. This matrix is designed to encode all the past input history, finding a map from the input to a higher dimensional space that represents the compression of the history. The matrix allows the model to be conceptualized as a convolutional model that decomposes an input signal onto an orthogonal system of smooth basis functions. In this way, the state encodes the history of the signal. Audio effects, to a different extent, present time dependencies crucial for accurately modeling them. The ability to track long dependencies in the input-output audio is beneficial in this context. In addition, The S4D model offers faster training times than classic RNNs and is defined by parameterizing its state matrix as a diagonal matrix. As RNNs, SSMs offer advantages due to their ability to capture dependencies in time-series data by utilizing internal states rather than solely relying on the input received at each iteration, such as a lengthy array of past input samples. This aspect is crucial when the goal is to achieve an implementation for live audio, which requires low input-output latency, where input refers to both audio and control parameters. We restrict the experimentation to relatively small networks with low computational complexity, allowing implementation on real-time consumer-grade digital audio systems. Experiments are based on models we have developed to exhibit minimal input-output latency, using small blocks of audio samples we typically find in computer-based audio processing dataflow as input.

The audio examples, dataset, source code, and trained models described in this paper are available online [1].

2.1 Architecture

The architecture of the artificial neural network used in this study is shown in Figure 1, and consists of fully connected (FC) layers and an S4D layer. The input is an array including the 64 most recent samples of the effect input signal $\boldsymbol{x_n}$. This choice is to give more context to the network but without introducing significant latency. The input is linearly projected with an FC layer to compute $\boldsymbol{u_n}$, which is in turn fed to the S4D layer to obtain $\boldsymbol{o_n}$. The S4D layer does not present nonlinearity; for this reason, an FC layer with a softsign activation function follows, obtaining $\boldsymbol{\hat{o}_n}$. An FC layer with one unit is the output layer.

The conditioning block indicated in Figure 1 with a dashed line can be placed at different network points. In particular, before and after the S4D layer. The first variation utilizes a single conditioning layer after the linear projection performed by the linear FC layer on the input. The second presents the conditioning layer after the FC layer with nonlinearity. Lastly, the third one places conditioning on both the previously mentioned positions. These experiments should reveal the optimal positioning of the conditioning layer for the network processing of the audio samples.

2.2 Conditioning Block

The types of conditioning blocks we compare in this study are shown in Figure 2. These may include the FiLM layer or the GA layer. As mentioned earlier, the FiLM method applies an affine transformation to the vector $\boldsymbol{p_n}$ representing the conditioning information. The conditioning vector is fed to an FC layer to produce another vector $\boldsymbol{\eta_n}$ having double the size of the conditioning vector. The output vector is split into two equally sized vectors, and the following operation is then applied to $\boldsymbol{p_n}$:

$$\boldsymbol{\hat{p}_n} = \boldsymbol{\eta_1 p_n} + \boldsymbol{\eta_0} \tag{1}$$

where η_1 and η_0 are the vectors obtained from the output of the FC layer. Following the projection, it is common to apply a function, such as ReLU or sigmoid, to determine

[1] https://github.com/RiccardoVib/Conditioning-Methods-for-Neural-Audio-Effects

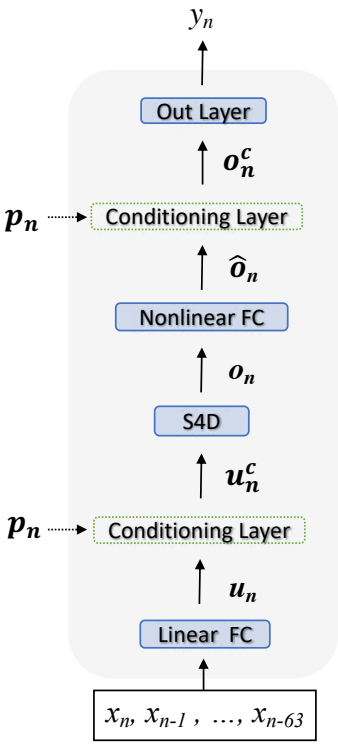


Figure 1. Architecture: the input is a vector including the 64 most recent input samples $[x_n, ..., x_{n-63}]$, which is fed to the network producing the output sample y_n. The network consists of a linear FC, S4D, and FC layer with softsign activation function adding the nonlinearity. The output layer is an FC layer with one unit. The dashed lines indicate the possible placement of conditioning layers: after the linear projection, just before the output layer, or in both positions.

the amount of information that should be passed. Our approach replaces the activation function with a Gated Linear Unit (GLU) [16]. This layer is more stable than ReLU and exhibits faster learning than sigmoid. Similarly to the previous step, the GLU layer consists of a linear FC layer that takes the FiLM output vector as input and computes a vector twice its length. The resulting output is split equally into two vectors: $\boldsymbol{q_1}$ and $\boldsymbol{q_2}$. A function is applied to $\boldsymbol{q_2}$, and the resulting output is multiplied element-wise with $\boldsymbol{q_1}$. The following equation describes the GLU layer:

$$\boldsymbol{o_n^c} = \boldsymbol{q_1} \otimes softsign(\boldsymbol{q_2}) \tag{2}$$

The GLU layer determines the flow of information through the network, acting as a logical gate. Typically, a sigmoid activation function is used for this purpose. However, in our approach, we introduce a softsign function instead. The softsign function controls the extent to which the control parameters should positively or negatively influence the final output. A value of 0 indicates a bypass, meaning no influence on the output. This allows for more flexibility.

The gated activation (GA), instead, is described by the following expression:

$$\boldsymbol{o_n^c} = tanh(\boldsymbol{h_n}) \otimes sigmoid(\boldsymbol{h_n}). \tag{3}$$

where

$$\boldsymbol{h_n} = \boldsymbol{x_n} * \boldsymbol{W_f} + \boldsymbol{p_n} * \boldsymbol{W_g} \tag{4}$$

with $\boldsymbol{W_f}$, $\boldsymbol{W_g}$ representing the learned convolution weights. Lastly, a modification of the GLU, named Gated Convolutional Unit (GCU), is proposed by adding two parallel convolutional layers instead of the linear FC layer. The primary objective of this study is to explore and extend the conditioning methods to investigate how effective they are in influencing inferring networks based on the conditioning information. For this reason, we applied and compared the previously mentioned methods, the GA, FiLM-GLU, and FiLM-GCU, placing them at different stages of the architecture (*post*, *pre*, *pre-post*). All the conditioning layers are designed to have a similar number of parameters. Finally, when using the FiLM layer, we experiment also with the following nonlinear transformations:

$$\begin{aligned} f(\boldsymbol{p_n}) &= \boldsymbol{\eta_1} \boldsymbol{p_n}^3 + \boldsymbol{\eta_0} \\ f(\boldsymbol{p_n}) &= \boldsymbol{\eta_1} \boldsymbol{p_n}^5 + \boldsymbol{\eta_0} \end{aligned} \tag{5}$$

2.3 Dataset

For this study, we selected two different audio effects: overdrive distortion and dynamic range compression. These types are representative of the effects already successfully modeled using black-box neural network techniques. For each effect unit, we selected only two variable control parameters. Our primary goal is to evaluate the prediction accuracy of the networks when influenced by control parameters used as conditioning information. We collected data from effects implemented as software plugins to have enough flexibility. The data were collected using a tool that facilitates and speeds up the collection of reproducible datasets from audio effects [2]. The parameters were sampled with 3 equally spaced values across the ranges.

The selected effects are FabFilter Pro-C 2 (C1) [3], SphereComp (C2) [4], Auditory Dr Drive (OD1) [5], Otto8 Overdrive (OD2) [6]. The selected variable parameters are the ratio and threshold for the compressor, drive, and tone levels for the overdrive. The ratio range is [1.5:1, 10:1], while the threshold is between [-3, -36] dB. The overdrive effect's parameters are percentages that do not relate to measurable quantities. The selected range is [10%, 100%]. For both effects, the selected range determines at least some compression or distortion (i.e., no settings generating a fully dry output signal). Table 1 summarizes the ranges and values for each audio effect. The dataset from the selected audio effects was developed by recording the effect's output for different combinations of the selected control parameters. Three equally spaced values were recorded for each parameter, resulting in 9 combinations for each effect. The audio data was recorded at a

[2] https://github.com/stefanofasciani/DGMD
[3] https://www.fabfilter.com/products/pro-c-2-compressor-plug-in
[4] https://www.waproduction.com/plugins/view/spherecomp
[5] https://www.audiority.com/shop/dr-drive/
[6] https://www.pluginboutique.com/products/8864-OTTO8-OVERDRIVE

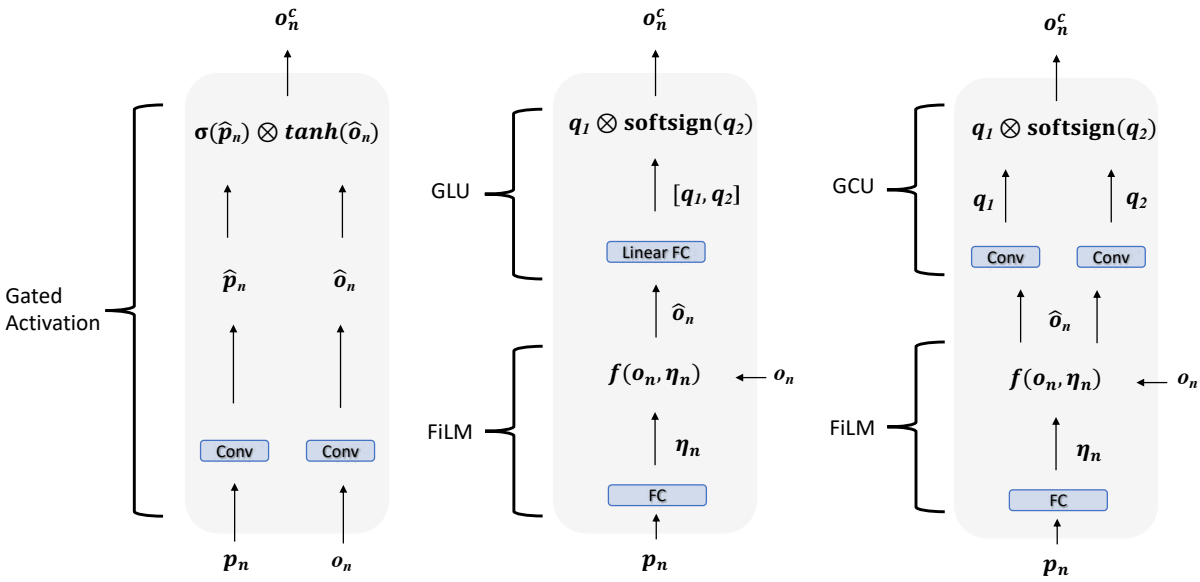


Figure 2. Conditioning layers. Gated Activation (left): the vector representing the conditioning information $\boldsymbol{p_n}$ and the input vector $\boldsymbol{o_n}$ are passed to two convolutional layers which to the outputs are applied the sigmoid function and the hyperbolic tangent, separately. Their results are element-wise multiplied together. FiLM-GLU (middle): the vector representing the conditioning information $\boldsymbol{p_n}$ fed an FC. The outcome is the vector with a length double the input size. This output vector is then split into two vectors that are used to apply a transformation (Equation 1 or 5) indicated with $f()$ to the vector $\boldsymbol{o_n}$. Subsequently, the transformed vector passes through another linear FC layer, producing another output vector double the length of its input. This output is split into two halves: the first one is applied to the softsign function, and the second one is multiplied element-wise. FiLM-GCU (right): similar to the FiLM-GLU case, but after the transformation $f()$, the vector is passed to two convolutional layers.

Software	Parameters	Range
Compressors	Ratio	[1.5:1, 4:1, 10:1]
	Threshold	[-3, -20, -36] dB
Overdrives	Drive	[10%, 45%, 100%]
	Bright/Tone	[10%, 45%, 100%]

Table 1. Software effects selected for the experiments, with parameters and their respective ranges.

sampling rate of 48 kHz. For each combination, the effects were fed with a mono input signal with a duration of 2 minutes. The input signal includes a variety of sounds, such as frequency sweeps covering a range of 20 Hz to 20 kHz, white noises with increasing amplitudes (both linear and logarithmic), recordings of instruments such as guitar, bass, drums (both loops and single notes), vocals, piano, pad sounds, and sections from various electronic and rock songs. The control parameters of each effect are mapped to the range $[0, 1]$ before using them as network conditioning information.

The C1 and OD1 datasets were used for the experiments investigating the conditioning methods and the placements, while the latter two were employed for further experiments aiming to confirm the findings of the previous experiments.

2.4 Experimenting and Learning

The models are trained for 60 epochs and use the Adam [17] optimizer with a gradient norm scaling of 1 [18]. The training was stopped earlier in case of no reduction of validation loss for 10 epochs. We design a time-based schedule for the learning rate as follows:

$$lr = \frac{LR}{e^4} \tag{6}$$

with LR the initial learning rate, that in our case is $3 \cdot 10^{-4}$ and e the number of the current epoch. The loss function used is the mean square error (MSE) and is computed using the model's weights that minimize the validation loss throughout the training epochs. The input signal is split into segments of 600 samples (equivalent to 12.5 ms) to be processed before updating the weights. The models are evaluated using the MSE, error-to-signal-ratio (ESR), and multi-resolution short-time Fourier transform (STFT) error with resolutions of [256, 512, 1024] samples.

Finally, the dataset is split into 80% for the training set, 10% for the validation set, and the remaining 10% for the test set. The data was divided into an $80 - 10 - 10\%$ split at the individual recording level, ensuring that all control parameter combinations were equally represented in each subset and that they were used consistently for both training and evaluating the model. The test set includes audio unseen during training, while all parameter combinations are included in the training set. We do not evaluate the

models' ability to predict audio for conditioning parameters never seen during training. Minor manual adjustments are made to ensure that splitting points for audio recordings fall within segments of silence.

3. RESULTS

Tables 2 and 3 report the prediction errors related to different types of conditioning layers and placements. These errors relate to models trained using the C1 and OD1 datasets. In both cases, the baseline model, as expected, shows lower performances. When considering the compressor, in Table 2, the GA method presents larger errors in all cases. The affine transformation of the FiLM method results is beneficial, especially if placed after the S4D layer, and even if with smaller differences, using fully-connected layers in the gated unit leads to generally better prediction accuracy. These performances can also be observed in Figure 3, where the compression curves of the target and the model's predictions are shown. In particular, the curves refer to the three different ratios against a threshold set to -20 dB. The FiLM layer shows the best performance, especially when encountering heavier compression. On the other hand, GA particularly underestimates the compression in all the cases. Conversely, for the overdrive, the GA performs generally better than when applied to the compressor. In addition, using convolution in the gated unit after the FiLM operation leads to the best results in all cases, indicating convolution is particularly beneficial in overdrive modeling. The best accuracy is found in the FiLM-GCU case when placed before the S4D layer, but it presents a small gap with the *post* placements. In addition, placing the conditioning after the S4D layer led to general improvements when considering the other methods. This suggests that also, in the overdrive case, it is more beneficial for the networks to use the information given by the control parameters to project the output of the S4D layer and determine the extent to which this information influences the final output rather than influencing the inference of the recurrent layer based on the control parameters. Lastly, we note that the *pre-post* configuration leads to the worst performance for the overdrive dataset. From Figure 4, we observe that the three methods perform similarly, predicting the conditioning case in the middle of the range. GA improves when encountering heavy distortion and degenerates in the light distortion case, while FiLM shows the opposite trend. FiLM-based layers showed the best overall modeling accuracy in both cases, especially if placed after the S4D layer. GLU and GCU performance depends on the dataset. The overdrive type works better when convolutional layers are used, while the compressor shows better accuracy using a linear FC layer. The same trend is evident when training models with C2 and OD2 datasets, as visible in Table 4. These experiments are performed only in the *post* configuration. Again, convolution-based conditioning resulted in more benefits for the overdrive, while GLU remains a better candidate for compressors. The choice may depend on the influence of the various parameters on the output sound, particularly their non-linear relationship between control parameters and their impact on the effect's sound alteration process. To further investigate this aspect, we extend the transformation inside the FiLM layer with equations of order greater than one and odd. Even orders are discarded because of their symmetrical nature. The applied transformations are listed in Equation 5. Table 5 reports the errors, and as expected, in the case of the overdrive, a nonlinear transformation of or-

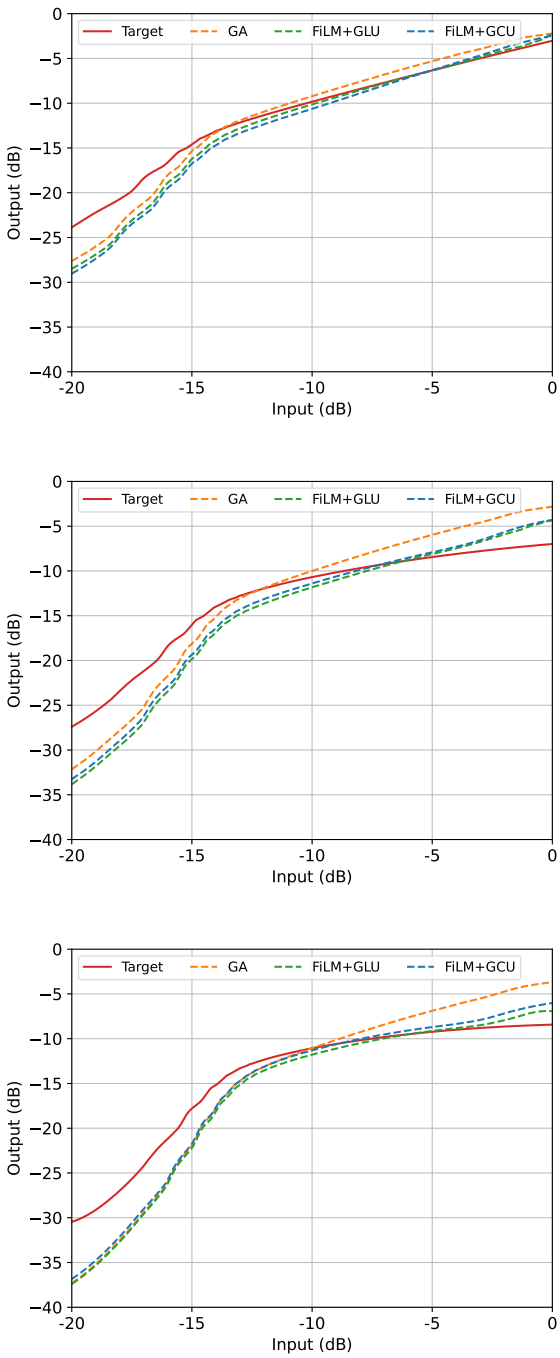


Figure 3. Compression curves of targets and predictions using the three different conditioning methods: GA, FiLM-GLU, and FiLM-GCU. The set threshold is -20 dB, while the ratio is 1.5:1 (top), 4:1 (middle), and 10:1 (bottom). The target example was not present in the training set. The plots refer to the C1 dataset.

Method	Stage	MSE	ESR	STFT
Baseline	-	$8.27 \cdot 10^{-3}$	$2.98 \cdot 10^{-1}$	$6.84 \cdot 10^{-1}$
GA	Post	$7.23 \cdot 10^{-3}$	$2.60 \cdot 10^{-1}$	$5.51 \cdot 10^{-1}$
FiLM-GLU	**Post**	$\mathbf{2.00 \cdot 10^{-3}}$	$\mathbf{7.24 \cdot 10^{-2}}$	$\mathbf{1.82 \cdot 10^{-1}}$
FiLM-GCU	**Post**	$2.70 \cdot 10^{-3}$	$9.73 \cdot 10^{-2}$	$3.15 \cdot 10^{-1}$
GA	Pre	$7.92 \cdot 10^{-3}$	$2.85 \cdot 10^{-1}$	$8.41 \cdot 10^{-1}$
FiLM-GLU	Pre	$4.49 \cdot 10^{-3}$	$1.62 \cdot 10^{-1}$	$5.02 \cdot 10^{-1}$
FiLM-GCU	Pre	$\mathbf{4.42 \cdot 10^{-3}}$	$\mathbf{1.59 \cdot 10^{-1}}$	$\mathbf{4.81 \cdot 10^{-1}}$
GA	**Pre-Post**	$6.81 \cdot 10^{-3}$	$2.45 \cdot 10^{-1}$	$7.34 \cdot 10^{-1}$
FiLM-GLU	Pre-Post	$\mathbf{3.37 \cdot 10^{-3}}$	$\mathbf{1.21 \cdot 10^{-1}}$	$\mathbf{4.00 \cdot 10^{-1}}$
FiLM-GCU	Pre-Post	$5.51 \cdot 10^{-3}$	$1.98 \cdot 10^{-1}$	$6.23 \cdot 10^{-1}$

Table 2. MSE, ESR, and STFT errors referring to the different conditioning methods and placements, if applicable, and for the compressor C1 case. In bold are indicated the best results for each method and placement.

Method	Stage	MSE	ESR	STFT
Baseline	-	$2.99 \cdot 10^{-2}$	$4.59 \cdot 10^{-1}$	$7.17 \cdot 10^{-1}$
GA	**Post**	$7.41 \cdot 10^{-3}$	$1.13 \cdot 10^{-1}$	$3.79 \cdot 10^{-1}$
FiLM-GLU	**Post**	$7.92 \cdot 10^{-3}$	$1.21 \cdot 10^{-1}$	$\mathbf{3.13 \cdot 10^{-1}}$
FiLM-GCU	Post	$\mathbf{7.17 \cdot 10^{-3}}$	$\mathbf{1.10 \cdot 10^{-1}}$	$3.31 \cdot 10^{-1}$
GA	Pre	$7.93 \cdot 10^{-3}$	$1.21 \cdot 10^{-1}$	$4.01 \cdot 10^{-1}$
FiLM-GLU	Pre	$8.24 \cdot 10^{-3}$	$1.26 \cdot 10^{-1}$	$3.77 \cdot 10^{-1}$
FiLM-GCU	**Pre**	$\mathbf{6.86 \cdot 10^{-3}}$	$\mathbf{1.05 \cdot 10^{-1}}$	$\mathbf{3.28 \cdot 10^{-1}}$
GA	Pre-Post	$9.24 \cdot 10^{-3}$	$1.41 \cdot 10^{-1}$	$4.05 \cdot 10^{-1}$
FiLM+GLU	Pre-Post	$8.39 \cdot 10^{-3}$	$1.28 \cdot 10^{-1}$	$3.97 \cdot 10^{-1}$
FiLM-GCU	Pre-Post	$\mathbf{8.00 \cdot 10^{-3}}$	$\mathbf{1.22 \cdot 10^{-2}}$	$\mathbf{3.65 \cdot 10^{-1}}$

Table 3. MSE, ESR, and STFT errors referring to the different conditioning layers and placements, if applicable, and for the overdrive OD1 case. In bold are indicated the best results for each method and placement.

Dataset	Method	Stage	MSE	ESR	STFT
C2	GA	Post	$8.31 \cdot 10^{-4}$	$8.71 \cdot 10^{-2}$	$3.49 \cdot 10^{-1}$
	FiLM-GLU	Post	$\mathbf{6.72 \cdot 10^{-4}}$	$\mathbf{7.04 \cdot 10^{-2}}$	$2.76 \cdot 10^{-1}$
	FiLM-GCU	Post	$7.11 \cdot 10^{-4}$	$7.45 \cdot 10^{-2}$	$\mathbf{2.75 \cdot 10^{-1}}$
OD2	GA	Post	$1.82 \cdot 10^{-3}$	$1.87 \cdot 10^{-1}$	$\mathbf{3.92 \cdot 10^{-1}}$
	FiLM-GLU	Post	$2.01 \cdot 10^{-3}$	$2.06 \cdot 10^{-1}$	$4.46 \cdot 10^{-1}$
	FiLM-GCU	Post	$\mathbf{1.71 \cdot 10^{-3}}$	$\mathbf{1.76 \cdot 10^{-1}}$	$4.06 \cdot 10^{-1}$

Table 4. MSE, ESR, and STFT errors referring to the C2 and OD2 datasets, with the different conditioning layers and *post* configuration. In bold are indicated the best results for each method and placement.

ders 3 and 5 results beneficial. A nonlinear function more accurately represents the distortion introduced as a function of the two parameters. However, when an equation of the fifth order in the opposite degenerates the performances in the model trained with the OD1 dataset, a fifth-order transformation may provide excessive steepness in the response of the conditioning layer for the OD1. Still, the behavior is the opposite for the OD2 dataset, which improves when increasing the order. Differently, the compressors do not benefit from nonlinear transformations, which is consistent with the previous results where GLU showed the best modeling accuracy.

4. CONCLUSION

Artificial neural networks have been used to model various audio effects, showing sufficient accuracy in most cases. At the same time, real-time and low-latency capability is still a challenge as many proposed models use large artificial neural networks. Most models are also limited in emulating the variable response of an audio effect, which generally occurs when tuning the parameters exposed to users. How to feed parameter-related information into the modeling networks is fundamental to achieve an accurate fully-conditioned emulation. In this study, we investigated this aspect, starting from an architecture based on a state-space layer. We compared two conditioning methods, the first based on gated activation and the other on feature-wise linear modulation. The latter is commonly used for

Dataset	Method	Order	MSE	ESR	STFT
C1	FiLM-GLU	**1**	$\mathbf{2.00 \cdot 10^{-3}}$	$\mathbf{7.24 \cdot 10^{-2}}$	$\mathbf{1.82 \cdot 10^{-1}}$
-	FiLM-GLU	3	$2.21 \cdot 10^{-3}$	$7.99 \cdot 10^{-2}$	$2.15 \cdot 10^{-1}$
-	FiLM-GLU	5	$2.51 \cdot 10^{-3}$	$9.05 \cdot 10^{-2}$	$2.64 \cdot 10^{-1}$
-	FiLM-GCU	**1**	$\mathbf{2.70 \cdot 10^{-3}}$	$\mathbf{9.73 \cdot 10^{-2}}$	$\mathbf{3.15 \cdot 10^{-1}}$
-	FiLM-GCU	3	$3.98 \cdot 10^{-3}$	$1.43 \cdot 10^{-1}$	$4.65 \cdot 10^{-1}$
-	FiLM-GCU	5	$5.20 \cdot 10^{-3}$	$1.87 \cdot 10^{-1}$	$4.44 \cdot 10^{-1}$
C2	FiLM-GLU	**1**	$\mathbf{6.72 \cdot 10^{-4}}$	$\mathbf{7.04 \cdot 10^{-2}}$	$\mathbf{2.76 \cdot 10^{-1}}$
-	FiLM-GLU	3	$8.88 \cdot 10^{-4}$	$9.31 \cdot 10^{-2}$	$3.39 \cdot 10^{-1}$
-	FiLM-GLU	5	$8.69 \cdot 10^{-4}$	$9.11 \cdot 10^{-2}$	$3.40 \cdot 10^{-1}$
-	FiLM-GCU	**1**	$\mathbf{7.11 \cdot 10^{-4}}$	$\mathbf{7.45 \cdot 10^{-2}}$	$\mathbf{2.76 \cdot 10^{-1}}$
-	FiLM-GCU	3	$7.77 \cdot 10^{-4}$	$8.15 \cdot 10^{-2}$	$3.03 \cdot 10^{-1}$
-	FiLM-GCU	5	$9.19 \cdot 10^{-4}$	$9.64 \cdot 10^{-2}$	$3.60 \cdot 10^{-1}$
OD1	FiLM-GLU	1	$7.92 \cdot 10^{-3}$	$1.21 \cdot 10^{-1}$	$3.13 \cdot 10^{-1}$
-	FiLM-GLU	**3**	$\mathbf{7.32 \cdot 10^{-3}}$	$\mathbf{1.12 \cdot 10^{-1}}$	$\mathbf{3.10 \cdot 10^{-1}}$
-	FiLM-GLU	5	$9.08 \cdot 10^{-3}$	$1.39 \cdot 10^{-1}$	$3.23 \cdot 10^{-1}$
-	FiLM-GCU	**1**	$\mathbf{7.17 \cdot 10^{-3}}$	$\mathbf{1.10 \cdot 10^{-1}}$	$\mathbf{3.31 \cdot 10^{-1}}$
-	FiLM-GCU	3	$7.57 \cdot 10^{-3}$	$1.16 \cdot 10^{-1}$	$3.55 \cdot 10^{-1}$
-	FiLM-GCU	5	$8.96 \cdot 10^{-3}$	$1.23 \cdot 10^{-1}$	$3.67 \cdot 10^{-1}$
OD2	FiLM-GLU	1	$2.01 \cdot 10^{-3}$	$2.06 \cdot 10^{-1}$	$4.46 \cdot 10^{-1}$
-	FiLM-GLU	**3**	$\mathbf{1.30 \cdot 10^{-3}}$	$\mathbf{1.34 \cdot 10^{-1}}$	$\mathbf{3.07 \cdot 10^{-1}}$
-	FiLM-GLU	5	$1.38 \cdot 10^{-3}$	$1.42 \cdot 10^{-1}$	$3.24 \cdot 10^{-1}$
-	FiLM-GCU	1	$1.71 \cdot 10^{-3}$	$1.76 \cdot 10^{-1}$	$4.06 \cdot 10^{-1}$
-	FiLM-GCU	3	$1.28 \cdot 10^{-3}$	$1.31 \cdot 10^{-1}$	$3.07 \cdot 10^{-1}$
-	FiLM-GCU	**5**	$\mathbf{1.19 \cdot 10^{-3}}$	$\mathbf{1.22 \cdot 10^{-1}}$	$\mathbf{3.05 \cdot 10^{-1}}$

Table 5. MSE, ESR, and STFT errors referring to FiLM-GLU and FiLM-GCU conditioning methods, with different transformations as listed in Equation 5. In all cases, the conditioning layer is placed after the S4D layer.

conditioning neural networks in other application domains. The technique includes performing an affine transformation via learnable coefficients based on conditioning information. We improved the methods by adding gated linear and gate convolutional units. The study is based on two types of audio effects, distortion, and compression, which have proven to be accurately modeled using artificial neural networks. Distortion and compression have been chosen due to their significantly different sound-altering characteristic. Results showed the FiLM outperforms other approaches, and its placement after the S4D layer leads to more accurate emulations of the audio effect response. Both are compared when placed at different stages of the architecture and used as baseline feeding the conditioning information concatenated to the input samples. The gated linear unit is beneficial in the compressor case, while the convolutional counterpart is more suitable for the overdrive. In addition, the FiLM layer considering nonlinear transformations is beneficial for the overdrive effect, which parameters may present a stronger nonlinear relationship with the associated sonic response of the effect. In particular, a nonlinear transformation of the third order led to better results for the overdrive dataset, while no improvements were noted for the compressor.

The study will be extended to various effects in future work, incorporating control parameters with different characteristics. In particular, this study does not include parameters changing the temporal profile of the compression, namely the attack and release time. In addition, nonlinear mappings in the conditioning layer have been found beneficial when modeling overdrive. Future studies will be extended to include more refined transformations.

5. REFERENCES

[1] D. S. Mendoza, *Emulating electric guitar effects with neural networks*. Barcelona, Spain: Universitat Pompeu Fabra, 2005.

[2] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep learning for tube amplifier emulation," in *Proc. 2019 Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 471–475.

[3] E.-P. Damskägg, L. Juvela, V. Välimäki *et al.*, "Real-time modeling of audio distortion circuits with deep learning," in *Proc. of 16th Int. Sound and Music Computing Conf. (SMC)*, Malaga, Spain, 2019.

[4] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *Speech Synthesis Workshop*.

[5] J. Covert and D. L. Livingston, "A vacuum-tube guitar amplifier model using a recurrent neural network," in *Proc. of 2013 IEEE Southeastcon*, Jacksonville, Florida, USA, 2013.

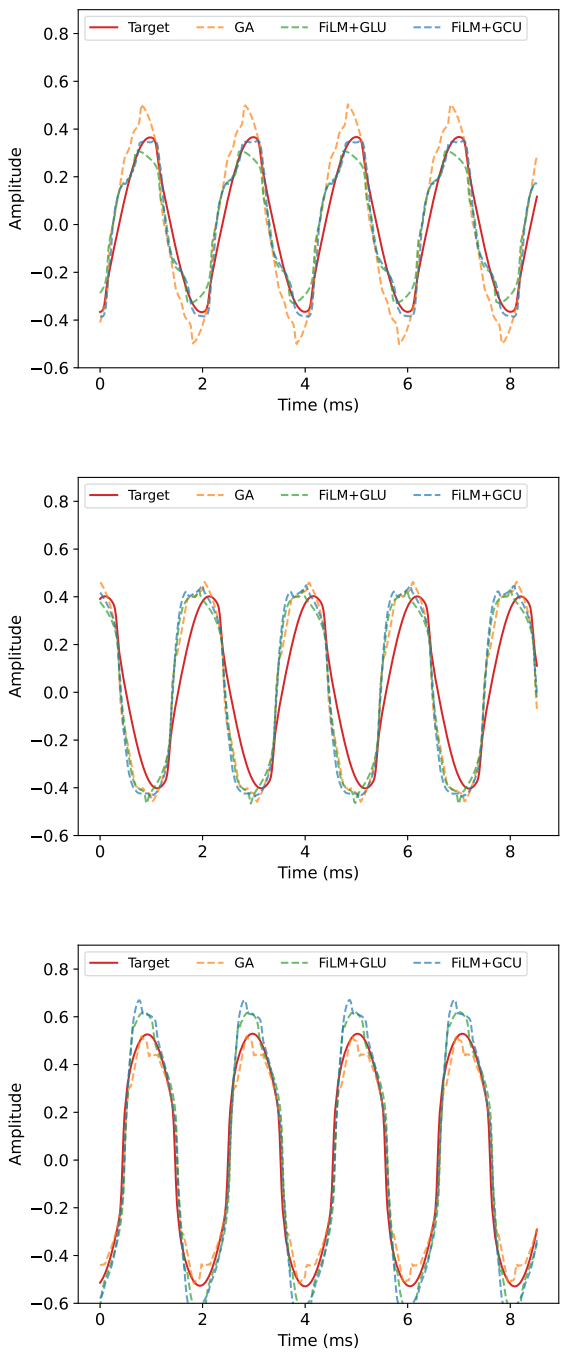


Figure 4. Target distorted 1000 Hz sine wave against the respective prediction using the three different conditioning methods: GA, FiLM-GLU, and FiLM-GCU. The drive knob is set to 45%, while the bright knob is to 10% (top), 45% (middle), and 100% (bottom). The sine wave is not present in the training set. The plots refer to models trained using the OD1 dataset.

[6] A. Wright, E.-P. Damskägg, and V. Valimaki, "Real-time black-box modelling with recurrent neural networks," in *Proc. of 22nd Int. Conf. Digital Audio Effects (DAFx)*, Birmingham, UK, 2019.

[7] J. Chowdhury, "A comparison of virtual analog modelling techniques for desktop and embedded implementations," *arXiv preprint arXiv:2009.02833*, 2020.

[8] R. Simionato and S. Fasciani, "Deep learning conditioned modeling of optical compression," in *Proc. of 25th Int. Conf. Digital Audio Effects (DAFx)*, Vienna, Austria, 2022.

[9] ——, "Fully conditioned and low-latency black-box modeling of analog compression," in *Proc. of 26th Int. Conf. Digital Audio Effects (DAFx)*, Copenaghen, Denmark, 2023.

[10] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time analog audio effect modeling," in *152nd Audio Engineering Society Convention*, The Hague, Netherlands, 2022.

[11] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. of 2018 Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, 2018.

[12] A. Gu, K. Goel, and C. Ré, "Efficiently modeling long sequences with structured state spaces," *Proc. of 2022 Int. Conf. Learning Representations (ICLR)*, 2022.

[13] A. Gu, K. Goel, A. Gupta, and C. Ré, "On the parameterization and initialization of diagonal state space models," *Proc. of 36th Int. Conf. Neural Information Processing Systems (NIPS)*, 2022.

[14] R. Simionato and S. Fasciani, "Comparative study of recurrent neural networks for virtual analog audio effects modeling," 2024.

[15] A. Gu, T. Dao, S. Ermon, A. Rudra, and C. Ré, "Hippo: Recurrent memory with optimal polynomial projections," *Proc. of 34th Int. Conf. Neural Information Processing Systems (NIPS)*, 2020.

[16] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, Sydney, Australia, 2017.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. of 2015 Int. Conf. Learning Representations (ICLR)*, 2015.

[18] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, Atlanta, USA, 2013.

PROGRAM NOTES

CONCERT 3

/ Les espaces negatives

/ The Voices

/ ‘Rust’

LES ESPACES NEGATIFS

Pierre Alexandre Tremblay
University of Huddersfield

I love a garden as a composed space, an aesthe:c simplifica:on of a more complex nature. The trace of the gardener is, beyond what is presented, in what is le@ out, to highlight a perspec:ve, to dis:nguish a figure over a background. A judgement.
Thanks to the CIRMMT team for the research residencies (Fabrice, Julien, Yves, Sylvain, Jackie, Bob, Philippe-Aubert, Dominic) and to my four co-explorers of such facili:es (Yulin, Philippe, Kasey, Nicola). Thanks to the CeReNeM and to FluCoMa for the financial support, to the Pohu-Traube family for the hospitality.

THE VOICES

Ji Youn Kang

"The Voices" is a performance as well as an intricate exploration of sound spatialization within the realm of live electronic music. This piece, featuring two microphones, eight channel loudspeakers, and a performer, delves into the complexities and challenges of creating spatial gestures during live performance. The introduction of stabilised audio feedback, created by the distance between the two microphones on the wrists of the performer and the eight loudspeakers, serves as a primary sound source, engaging with the space's resonances and revealing its unpredictable acoustics as a pivotal musical component. Thus, the piece evolves from a singular tone into a complex sonic structure through both movements of the performer and an elaborate network of processing modules that interact internally and circulate continuously, transforming "The Voices" into a living, breathing entity that communicates beyond the realm of words.
The dynamics of the piece interplay between sound and its spatial manifestations, where cohesion is meticulously crafted through spatialization of sound and the gestures of the performer in harmony with the Controlled-Ecosystemics system. Inspired by Agustino Di Scipio's principles of Ecosystemics, the Controlled-Ecosystemics system implemented for this piece approach employs microphones as 'moving ears,' enabling the performer to intuitively interact with the sound as it navigates the space. This setup fosters a captivating dialogue between the simplicity of the initial tones and the dynamic, spatial environment of the performance, with the performer's movements and responses enriching the auditory experience. invites the audience to immerse themselves in an auditory journey, engaging with sound in a profoundly immersive and reflective manner. Through this exploration, "The Voices" challenges and expands the boundaries of how sound can be generated, manipulated, and experienced, offering a unique opportunity to engage with the unseen forces that shape our sonic environment.
This piece was composed as a result of the lectorate of artistic research at The Royal Conservatoire in The Hague and advanced through the artistic research project 'Speculative Sound Synthesis,' hosted at the Institute for Electronic Music and Acoustics of the University of Music and Performing Arts Graz. It is funded by the Austrian Science Fund within the program for arts-based research (PEEK) – PEEK AR 713–G.

‘RUST’ FOR PIANO WITH SUBTLE ELECTRONIC

Luís Neto da Costa
Aristotle University of Thessaloniki

Rust is a piece for piano with subtle electronics, where the author explores the piano as a microtonal/ spectral instrument by preparing it with magnets and adhesive. The piano’s explorations are accompanied by a subtle layer of electronics. Sounds created through additive synthesis blend seamlessly with the piano’s natural resonance, creating a sense of sonic unity. There are six voices with different tempo markings and irrational time signatures based on the harmonic series. Each voice is imbued with a distinct rhythmic identity, forming a kind of polyphony, but also a main melody that explores different microtonal intervals and acoustic beatings. Because of this complex rhythmic notation, it would be humanly impossible to perform it as a paper score. Therefore, there is a video score with a fixed playback line while the score is moving, and the pianist has to play the note at the right moment. The piece showcases how rhythmic precision can be achieved through the use of new technological tools.

PAPER & POSTER /DEMO SESSION 7

/ Hybrid Neural Audio Effects

/ Implementation of a Continuously Variable Delay Line by Crossfading Between Several Tap Delays

/ Real-time Psychoacoustic Frequency Masking Compensation for Audio Signals with Overlapping Spectra

/ Modeling the Pultec EQP-1A with Wave Digital Filters

/ Revitalizing Multimedia Art: Restoration and Re-Activation of an Audiovisual Installation in Virtual Reality

/ Music Theatre and Digital Archives

/ Phase Repair for Time-Domain Convolutional Neural Networks in Music Super-Resolution

/ Flow: a Mutitimbral, Polyphonic, Modular, Additive Music Synthesizer

/ Loom: a Speculation of Oral Tradition in Immersive Media

/ Exploring the Space Between Instrument and Controller

HYBRID NEURAL AUDIO EFFECTS

Riccardo SIMIONATO (riccardo.simionato@imv.uio.no)[1] and
Stefano FASCIANI (stefano.fasciani@imv.uio.no)[1]

[1]*Department of Musicology*, **University of Oslo**, Norway

ABSTRACT

Machine learning techniques are commonly employed for modeling audio devices, particularly analog audio effects. Conditioned models have been proposed as well. The conditioning mechanism utilizes the control parameters of the modeled device to influence the sound alteration process. Neural networks are shown to be capable of interpolating between conditioning parameter values. This paper further investigates the interpolation ability of neural audio effects. In particular, we introduce additional conditioning parameters to instruct the neural network to learn and predict different audio effects using Feature-wise Linear Modulation and the Gated Linear Unit. The resulting model is a hybrid neural effect that can reproduce, depending on the conditioning values, the audio-altering process of a specific audio effect or interpolates between them. We created hybrid audio effects from a preamp circuit, an optical compressor, and a tape recorder. The designed models are able to learn the sound alteration processes for individual effects and their combinations without producing audible artifacts, and users can use the conditioning parameters to navigate in a continuous space where each point represents a different hybrid audio effect.

1. INTRODUCTION

Neural networks have been extensively used to model analog audio effects. Starting from a distortion pedal, using a multilayer forward network [1], and continuing with convolutional-based networks for vacuum-tube amplifiers and various distortion pedals [2, 3]. Recurrent Neural Networks (RNNs), such as Long-Short Time Memory (LSTM) and Gated recurrent unit (GRU) networks have been investigated as well for similar types of effects [4, 5]. Works including control parameters exploited gated activation function when using convolutional-based networks, while for the case of RNNs, the control parameter has been considered as an extra input to the network. Compressor audio effects have been explored as well. Temporal Convolution Network (TCN) has been used for an optical compressor [6] together with the Feature-wise Linear Modulation (FiLM) method [7] as a conditioning method. In this case, the network models also have two conditioning parameters of the target device. A sequence-to-sequence model, based on recurrent networks, was used for another optical compressor. In this case, the conditioning is incorporated by exploiting the LSTM internal states [8]. Starting with two control parameters, the work was later extended to a fully conditioned model, including all four control parameters of the target device [9]. Delay-based effects could present additional challenges due to the variable temporal misalignment between input and output. In [10], a tape-based delay is modeled using GRU networks. The delay trajectory is first analyzed and extracted using impulse train signals. This signal represents the variable delay in samples over time. It is used to demodulate the signal before training the models or to directly guide a delay line built using differentiable signal processing [11]. In this way, the network can learn the saturation of the magnetic tape and an arbitrary delay line can be added a second time. LFO delay-based effects have been considered as well. In [12], authors use as extra input the LFO signal representing the frequency response of a time-varying system over time. By doing so, the model can theoretically produce LFO delay-based effects with arbitrary modulations. Combining the interpolation capability of the neural networks and the possibility of creating hybrid analog audio effects with a non-existent equivalent in the real world, we conceived a multi-type effect combining three analog effect types: a vacuum tube-based preamplifier, an optical compressor, and a tape recorder. The conditioning process uses the FiLM method, while the model inference is based on RNNs. The design of a neural model that emulates various analog audio and interpolates among them has not yet been investigated in the literature. Here, we propose using neural networks to create a hybrid between different sound coloring audio effects. In particular, the proposed model emulates the saturation-based coloring of tape recorders, tube-based amplification, and optical-based compression. These devices are based on three different and distant physical mechanisms. The resulting models navigate a one- or multi-dimensional space between two or three sound colorings. The rest of the paper is organized as follows. Section 2 details the functioning of the effect considered in this work. Section 3. describes the methodology and dataset we collected for our studies. Section 4 reports and discusses the results, while Conclusions are included in Section 5.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. SELECTED ANALOG EFFECTS

The first effect we selected for this experimentation is the Universal Audio vacuum tube preamplifier 610-B [1]. In this case, the device includes the Universal Audio 6176 Channel Strip, which also features a transistor-based limiter amplifier 1176LN [2].

The second selected effect is the TUBE-TECH CL1B compressor [3]. This device is tube-based and optical, where the audio signal feeds a lighting element that, in turn, illuminates a light-sensitive resistor in the compression circuit. The resistance affects the compression circuit, determining how much and how quickly it attenuates the incoming audio signal. The CL 1B presents an output tube-based push-pull amplifier with variable gain, which is used to add harmonic distortion after the compression stage and not to limit the dynamics.

Finally, the third selected effect is the Akai 4000D open-reel tape recorder [4]. Tape recorder devices are based on magnetic tapes, which store the signal [13]. These devices typically consist of a recording amplifier, a recording head, a moving tape medium, a playback head, and a playback amplifier. The input signal first passes into the amplifier circuit and into an eventual stage equalization to compensate for the sound coloring introduced by magnetic tape. The signal is then routed to the tape record head and to the output, which is summed with its delayed replicas. The recording head produces a spatial magnetic field determined by both the recording head's properties and the input current's magnitude. The playback head restores the signal to electrical form, which process depends on the tape speed. The resulting delay effect is given by the position of the recording head on the tape loop and the distance from the playback head. In the last stage, another amplification is applied to the signal before the output. The tape movement speed is not perfectly constant due to small fluctuations produced by imperfections in the tape transport mechanics. These imperfections include cyclical components produced by the moving parts in the transport mechanism and stochastic behavior. These inconsistencies in the movement can be heard as small fluctuations in pitch. Magnetic tapes also generate noise due to the recording and playback processes.

3. METHODOLOGY

In this paper, we explore the interpolation capability offered by ANNs for modeling a hybrid audio effect based on the sound-altering process of different devices. For this experimentation, we have selected three significantly different analog audio effects, such as a vacuum tube pre-amplifier, an optical compressor, and a tape recorder. When training the system, we use extra input parameters to inform the neural network which of the three effects has generated the current input-output raw audio pairs. These parameters are used to condition the model during inference, allowing for generating a mixture of processing characteristics learned from the different effects, even when including intermediate conditioning settings not encountered during training. This allows the user to navigate between the alteration characteristics of the three analog effects. To evaluate to what extent the neural network can learn in such a context, we utilize two or three different audio effects to train the hybrid. After training the network, we verify whether the model can accurately replicate the sound alteration process of individual effects using the same conditioning input used during training and evaluate if varying the conditioning parameters to new values generates audible artifacts or spurious responses.

3.1 Architecture

The model is based on an encoder-decoder LSTM-based architecture as in [8]. RNNs offer advantages due to their ability to capture time dependencies in the data by utilizing internal states rather than solely relying on the input to generate each output prediction. This aspect is crucial when the goal is to achieve low input-to-output audio latency, which is essential in live interactive applications.

The network architecture used in this work is shown in Figure 1. It consists of two LSTM layers, a linear fully connected (FC) output layer, and the conditioning block. The network utilizes the 64 most recent input audio samples to generate one output audio sample. The input is split into 63 past samples $[x_n, ..., x_{n-63}]$ to feed the first LSTM layer. The input sample x_n at the current time step is first sent to the second LSTM layer. The first LSTM layer acts as an encoder and processes the 63 samples to compute the internal states $[h, c]$, which are used by the second LSTM layer. The latter uses the internal states to infer the output sample. The output is sent to the conditioning block, which is manipulated based on a continuous control parameter indicating the perceptual of the type of effect to use. A linear FC layer with one unit computes the output sample. The LSTM layers present 8 hidden units each, while the FC layers inside the conditioning block have 16 each. The architecture presents a maximum of 856 trainable parameters, depending on the design of the conditioning strategy.

3.2 Conditioning

The conditioning block, shown in Figure 2, consists of a Feature-wise Linear Modulation (FiLM) layer followed by a Gated Linear Unit (GLU) [14] layer. The FiLM layer applies an affine transformation to the vector p_n representing the conditioning information:

$$\hat{\boldsymbol{o}}_{\boldsymbol{n}} = \boldsymbol{\eta_a} o_n + \boldsymbol{\eta_b} \tag{1}$$

where η_a and η_b are two vectors obtained from splitting the output of a linear FC layer fed with p_n. A GLU layer follows the FiLM block to determine the amount of information that should be passed. The GLU layer consists of a linear FC layer that takes the FiLM output vector as input

[1] https://www.uaudio.com/hardware/mic-preamps/solo-610.html
[2] https://www.uaudio.com/hardware/6176.html
[3] http://www.tube-tech.com/cl-1b-opto-compressor/
[4] https://reel-reel.com/tape-recorder/akai-4000d/

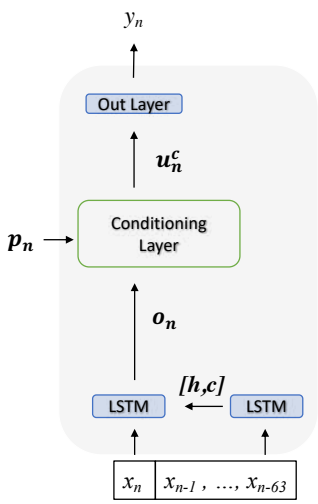


Figure 1. Architecture: the input is an array of the last 64 input audio samples $[x_n, ..., x_{n-63}]$, which is used to produce the output sample y_n. The input samples $[x_{n-1}, ..., x_{n-63}]$ are used as input of an LSTM layer, acting as an encoder. The input sample at the current time step x_n is first transformed by another LSTM layer acting as a decoder using internal states from the encoder. The output of the decoder LSTM layer is processed by the conditioning block followed by a linear FC layer with one unit, which produces the output audio sample y_n.

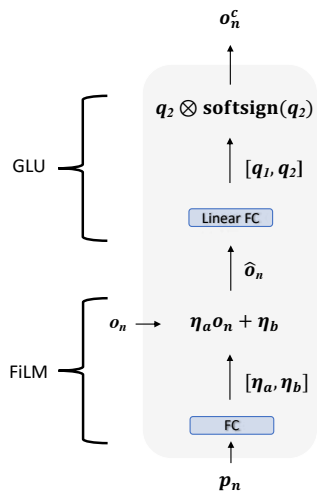


Figure 2. Conditioning Block: the conditioning information vector p_n fed a linear FC layer that produce an output with double its input length. This output vector is split into two vectors used to apply an affine transformation on the vector o_n, which represents the current input sample of the network. Subsequently, the transformed vector passes through another linear FC layer, producing another output vector double the length of its input. This output is also split into two halves. To one half is applied the softsign function, and the result is multiplied element-wise to produce the final output.

and computes a vector with twice its length. The resulting output is split equally into two vectors: q_1 and q_2. The softsign function is applied to q_2, and the resulting output is multiplied element-wise with q_1. Hence, the GLU block is described by the following:

$$\boldsymbol{o_n^c} = \boldsymbol{q_1} \otimes softsign(\boldsymbol{q_2}) \tag{2}$$

The GLU layer determines the flow of information through the network, acting as a logical gate. The softsign activation function controls the extent to which the control parameters should influence the final output.

3.3 Datasets

The microphone tube preamplifier datatset [5] was recorded from a 6176 Vintage Channel Strip unit [15]. The preamp was overdriven, setting the gain to +10dB, the output level to 6, and both high and low boost/cut to 0 dB. This configuration resulted in high harmonic distortion.

The compressor dataset [6] originally included 5 equal-spaced values for the threshold, ratio, attack, and release parameters. For this work, we limited to only the recording with the threshold set to -10 dB, ratio to 4:1, attack to 0.5 ms, and release 0.05 seconds.

Lastly, the tape delay dataset is from [10]. In particular, we used the MAXELL, $\frac{7}{2}$ inches per second configuration. Since the audio files in the dataset present time delay, they were also time-aligned using the delay trajectory, which was included in the datasets before the training.

We limited the datasets to 3 minutes and 20 seconds, which is the length of the shortest dataset, specifically the CL1B compressor dataset. By doing so, we ensure an equal amount of recordings from the two different effects in the datasets used for this study. In addition, 3 minutes at 44.1 kHz is shown as sufficient data for training [4]. We experimented with training 4 different hybrid neural audio effects. We utilize the three different pairs of datasets with a single conditioning parameter and the three datasets with a triplet of individual conditioning parameters, as detailed in Table 1. We also train three separate models with only one effect dataset as a baseline for modeling accuracy.

3.4 Experimenting and Learning

The models are trained for 60 epochs and use the Adam [16] optimizer with a gradient norm scaling of 1 [17]. The training was stopped earlier in case of no reduction of validation loss for 10 epochs. We design a time-based schedule for the learning rate as follows:

$$lr = LR * 0.25^e. \tag{3}$$

LR represents the initial learning rate, which is $3 \cdot 10^-4$, and e represents the current epoch's number. The loss function used is the mean square error (MSE) and is computed using the model's weights that minimize the validation loss throughout the training epochs. The input signal is split into segments of 600 samples (equivalent to 13.6

[5] https://zenodo.org/records/3562442
[6] https://www.kaggle.com/datasets/riccardosimionato/tubetech-cl-1b

Model	Dataset composition	p
P	preamplifier	-
C	compressor	-
T	tape recorder	-
TP	tape recorder	0.0
	preamplifier	1.0
CP	compressor	0.0
	preamplifier	1.0
CT	compressor	0.0
	tape recorder	1.0
CTP	compressor	$[1.0, 0.0, 0.0]$
	tape recorder	$[0.0, 1.0, 0.0]$
	preamplifier	$[0.0, 0.0, 1.0]$

Table 1. Models and datasets composition with the respective conditioning parameter values (p) used for the training.

Dataset	MSE	ESR	STFT
P	$4.34 \cdot 10^{-2}$	$1.55 \cdot 10^{-1}$	$3.94 \cdot 10^{-1}$
C	$1.52 \cdot 10^{-4}$	$1.07 \cdot 10^{-1}$	$3.10 \cdot 10^{-1}$
T	$3.46 \cdot 10^{-4}$	$1.44 \cdot 10^{-1}$	$2.51 \cdot 10^{-1}$
TP (T)	$1.36 \cdot 10^{-3}$	$5.66 \cdot 10^{-1}$	$8.11 \cdot 10^{-1}$
TP (P)	$5.05 \cdot 10^{-2}$	$1.81 \cdot 10^{-1}$	$3.83 \cdot 10^{-1}$
CP (C)	$4.11 \cdot 10^{-4}$	$2.91 \cdot 10^{-1}$	$5.42 \cdot 10^{-1}$
CP (P)	$5.00 \cdot 10^{-2}$	$1.79 \cdot 10^{-1}$	$3.78 \cdot 10^{-1}$
CT (C)	$1.03 \cdot 10^{-3}$	$7.31 \cdot 10^{-1}$	$9.63 \cdot 10^{-1}$
CT (T)	$4.52 \cdot 10^{-4}$	$1.88 \cdot 10^{-1}$	$3.02 \cdot 10^{-1}$
CTP (C)	$3.93 \cdot 10^{-4}$	$2.78 \cdot 10^{-1}$	$3.44 \cdot 10^{-1}$
CTP (T)	$2.16 \cdot 10^{-2}$	$9.02 \cdot 10^{-1}$	1.80
CTP (P)	$5.54 \cdot 10^{-2}$	$1.98 \cdot 10^{-1}$	$4.37 \cdot 10^{-1}$

Table 2. MSE, ESR, and STFT errors referring to the models trained detailed in Table 1. In this latter case, the errors also refer to the segment of the test set considering the conditioning cases separately. Specifically, T refers to the case of Tape, P to preamplifier, and C to compressor mode.

ms) to be processed before updating the weights. The models are evaluated using the MSE, ESR, and multi-resolution STFT with [256, 512, 1024] as resolutions.

Finally, the dataset is split into 80% for the training set, 10% for the validation set, and 10% for the test set. The $80 - 10 - 10\%$ split was carried out at the individual dataset level, which ensures that both effects examples are included with an equal share in each subset and are equally used for training and evaluating the model. Minor manual adjustments are made to ensure that splitting points fall within segments of silence.

4. RESULTS

Table 2 details the errors for all the cases. Specifically, when considering models trained with a mixture of audio effects, the error refers only to the segment of the test set representing a specific effect. In this way, we assert how much the model accuracy drops when including more effects compared to a model that includes only one of them. We can see how, as expected, all the errors particularly increased in the case of CTP. In CTP, we can note a drastic drop in performance for the tape recorder test set. CTP represents the most complex case, where a single model is expected to emulate significantly different audio alteration processes. Considering the case with two effects, the emulation of the tape recording when the preamplifier dataset is included is less accurate. At the same time, a minor drop in accuracy happens if the compression is learned together with the tape delay. Even if there is significantly less performance drop, the same happens with the preamplifier. Learning the compression is more difficult when it needs to be learned in combination with the tape recorder. This suggests that it is particularly challenging for the model to add and remove time misalignments between input and output and, in turn, the time shifting during the process. The TP model resulted in more challenges during the learning process, likely due to their effect on the sound: the preamplifier generates a significant harmonic distortion in the signal, while the tape recorder introduces time fluctuations. The compressor, on the other hand, requires less alteration of the signal. Looking at the spectrograms, as can be seen in Figure 3, the prediction of the CTP model, referring to the compressor, is the least affected by the mixture of effects, while the preamplifier and the tape recorder present more mismatch in the high-frequency content respect to the singular cases represented by the T and C model. In accordance with the STFT errors, the preamplifier is the least affected among the cases considering two effects, while the compressor presents more mismatch CTP, although the CTP model presents lower errors. The tape delay presents the most mismatch in frequency content in CTP. This could be because tape delay time shifting is present in a smaller percentage, considering the total amount of data. The time shifting in the TP and CP cases represents half of the seen examples. Additionally, as shown in Figure 4, the T and CT models similarly emulate the time misalignment introduced by the tape recorder dataset, while the TP model, modeling also the preamplifier, is less accurate with this aspect. Also, the TP model underpredicts the amplitudes. The CTP model, even if it still mispredicts the amplitude, better models the time fluctuations. On the other hand, all models still capture the time shifting. Figure 5 shows the predictions of CP, CT, and TP models when the parameter is set at the middle of the navigable space, such as 0.5. In this scenario, the model generates a hybrid effect between the two analog effects, representing situations that are not reproducible in the real-world and, in turn, are not comparable with real recordings. The plots show how the model creates the hybrid effects. For example, looking at the time plot, the CP model amplifies a compressed version of the input signal. Additionally, Figure 6 shows the spectrograms of the output models when continuously changing the conditioning parameters by modulating it using sine waves at 0.05 Hz. From the spectrogram, we can see that no visible artifacts were introduced in the process. Similarly, Figure 7 shows the predictions in the case of the CTP model considering various parameter values, such as $[0.3, 0.3, 0.3]$, $[0.5, 0.25, 0.25]$,

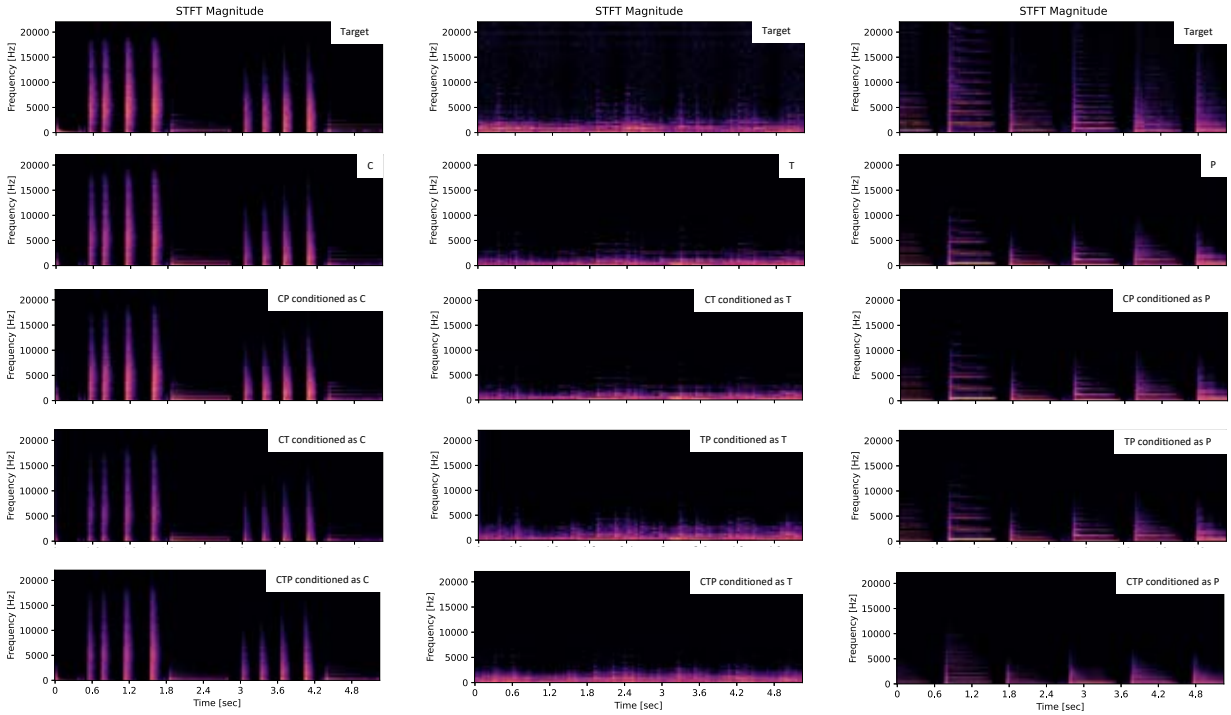


Figure 3. Normalized spectrograms of 5 seconds included in the test sets. Targets (top) against C (left), T (middle), and P (right) models, and against CT, TP, CP, and CTP models conditioned as C, as T, and as P, respectively.

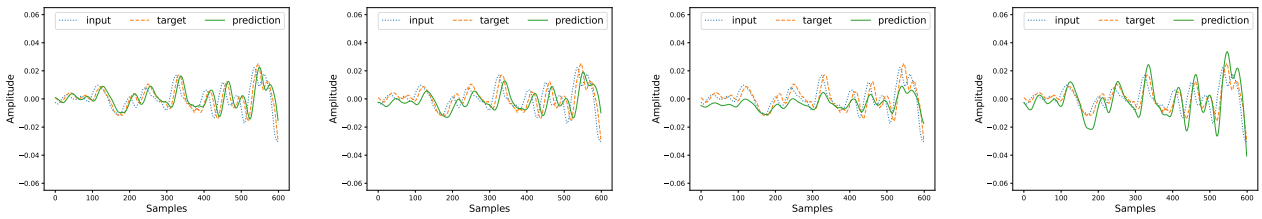


Figure 4. Input, target, and prediction referring to a segment of the tape recorder test set for the case of T, CT, TP, CTP, and TCP models. The plots show the time fluctuations present in the tape recorder dataset.

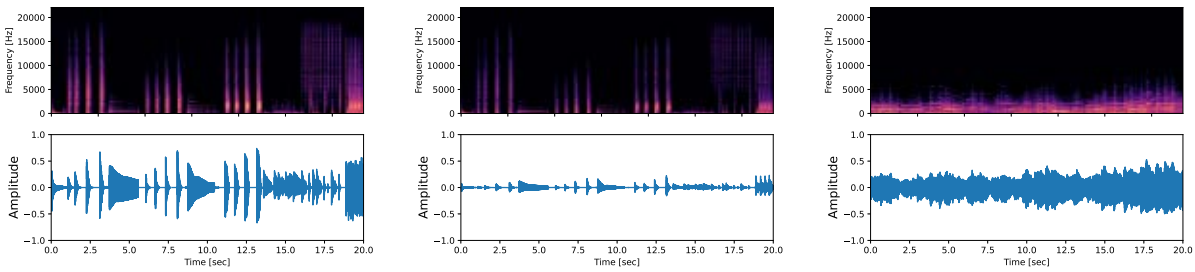


Figure 5. Waveforms and spectrograms (normalized) of 20 seconds included in the test sets, resulted from the CP (left), CT (middle), and TP (right) models when the parameter is 0.5.

$[0.25, 0.5, 0.25]$, and $[0.25, 0.25, 0.5]$. The conditioning vectors were chosen to have a norm not greater than 1.. On the other hand, in the CP, TP, and CT models, where the parameters switch continuously between the effects, the conditioning is designed differently in this case. In particular, CTP has 3 different parameters governing the amount of the effects to add to the input signal. With these conditioning values, all the waveforms present visible compression, enhanced in the case with $[0.5, 0.25, 0.25]$ as the conditioning vector, as expected. Analogously, when the conditioning vector is $[0.25, 0.25, 0.5]$, the model amplifies more than the other cases. To further explore the model behavior, Figure 8 presents extreme conditioning cases, such as $[0.0, 0.0, 0.0]$ and $[1.0, 1.0, 1.0]$. When the val-

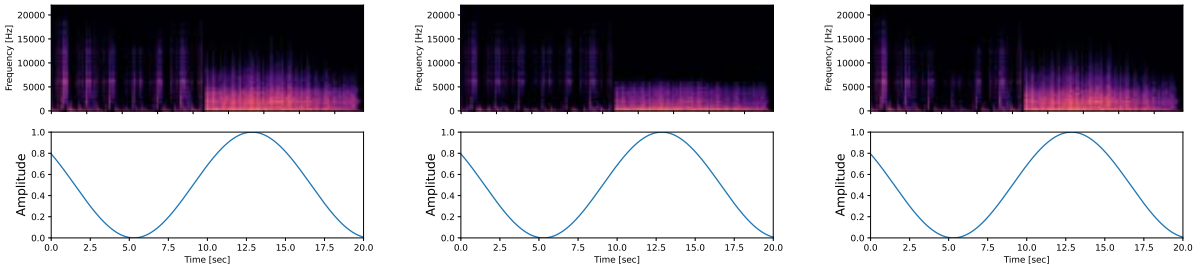


Figure 6. Normalized spectrograms of a 20 second signal resulted from the CP (left), CT (middle), and TP (right) models when the parameter is continuously changed by modulating it using a sine wave at 0.05 Hz.

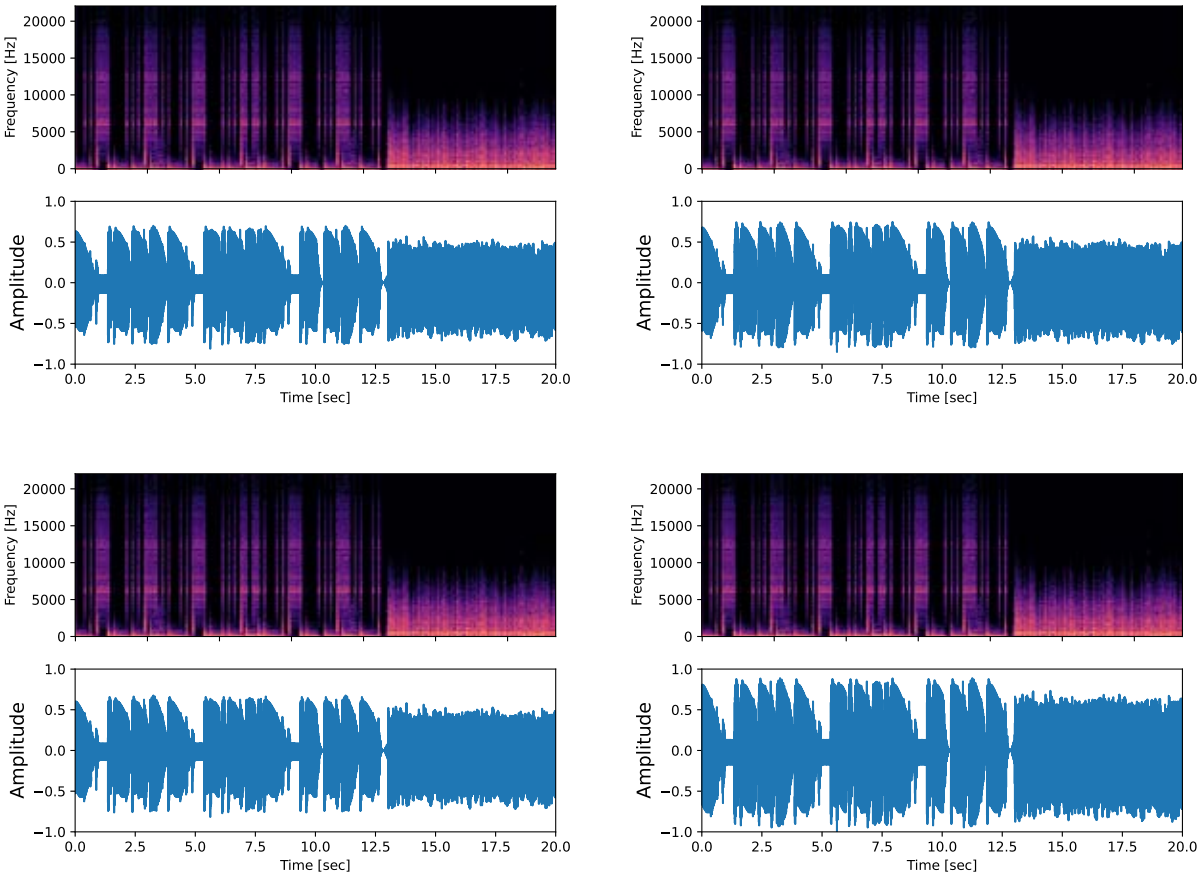


Figure 7. Waveforms and normalized spectrograms of a 20 second signal resulted from the CTP model when the parameter is $[0.3, 0.3, 0.3]$ (top-left), $[0.5, 0.25, 0.25]$ (top-right), $[0.25, 0.5, 0.25]$ (bottom-left), and $[0.25, 0.25, 0.5]$ (bottom-right).

ues are set to $[1.0, 1.0, 1.0]$, the model significantly amplifies the amplitude's input signal, as though three different predictions are added separately. Interestingly, when the parameter vector is set to $[0.0, 0.0, 0.0]$, the model significantly compresses the signal instead of replicating the input signal. This latter behavior suggests that the three parameters do not simply indicate the amount of the singular effects added to the input signal; instead, they describe a more complex transformation. Figure 8 shows two other cases with conditioning vectors having a norm greater than 1., such as $[0.5, 0.5, 0.5]$ and $[0.3, 0.8, 0.5]$. Also, in these cases, the signal is still significantly amplified and presents amplitudes greater than 1.. Finally, Figure 9 shows the spectrogram of the CTP model when continuously changing the conditioning parameters by modulating it using three sine waves at 0.05, 0.07, and 0.09 Hz. The figure presents the case of musical and sawtooth waves. Also here, as in all of the presented examples, as can be seen, do not introduce visible artifacts, nor when listening to the audio sample that can be found in [7].

[7] https://github.com/RiccardoVib/Hybrid-Neural-Audio-Effects

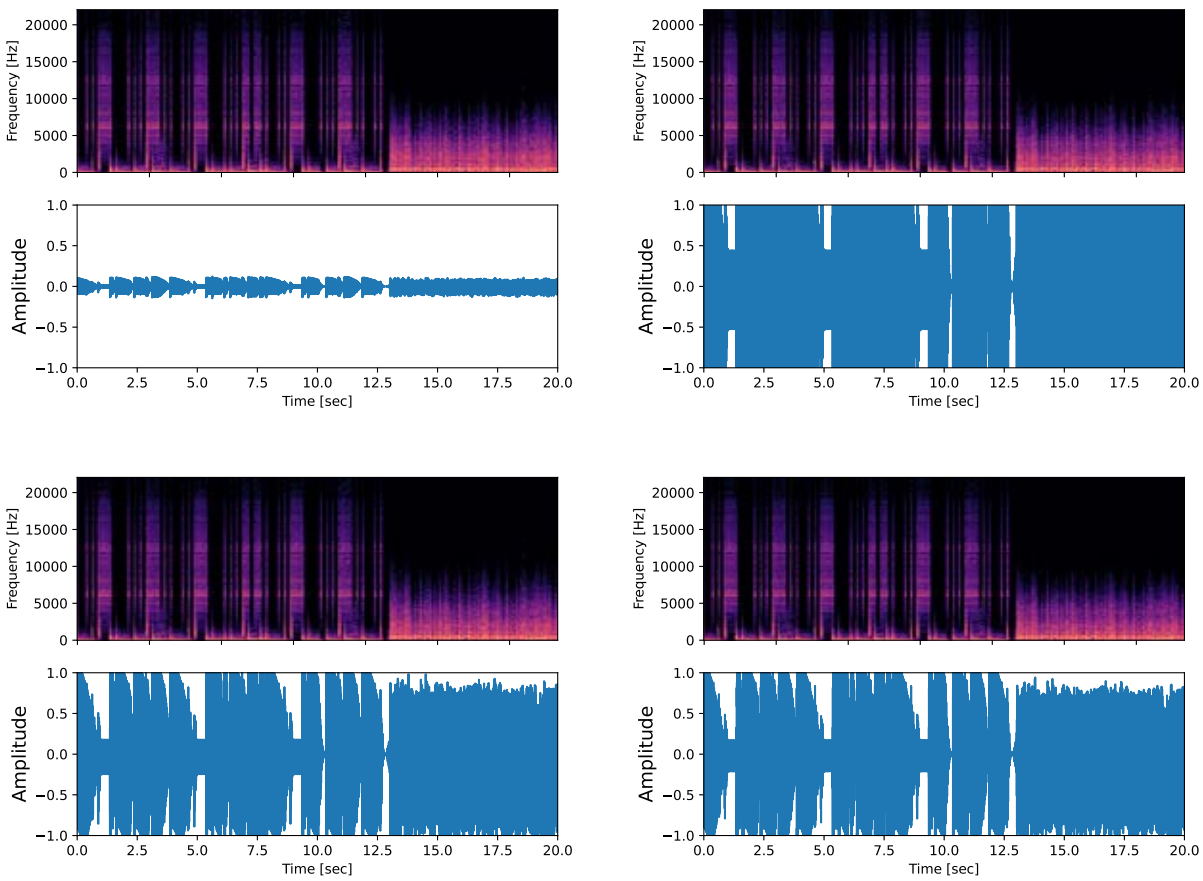


Figure 8. Waveforms and normalized spectrograms of a 20 second signal resulted from the CTP model when the parameter is [0.0,0.0,0.0] (top-left), [1.0,1.0,1.0] (top-right), [0.5,0.5,0.5] (bottom-left), and [0.3,0.8,0.5] (bottom-right).

5. CONCLUSION

Machine learning nowadays integrates digital signal processing to model physical systems, such as analog audio effects. In this context, the models must be conditioned based on the user control parameters to control how to alter the sound modification process. Neural networks are capable of interpolating between seen data. This paper investigates this interpolating ability to create hybrid models mixing different audio effects and navigating a space not directly existing in the physical world. With this goal, we defined continuous parameters with the range [0.0, 1.0] to condition the network and emulate different analog effects, such as a compressor, a preamplifier, and a tape recorder. We used three datasets from a real audio device and experimented with 4 different hybrid neural effects. Three were trained utilizing different pairs of datasets with a single conditioning parameter, and one was trained using all datasets conditioned with a three-dimensional vector. The conditioning layer is based on Feature-wise Linear Modulation and Gated Linear Unit. The final models can navigate between amplification, compression, and saturation types of effect, although with less accuracy than the model trained for a specific effect only. The hybrid models do not introduce visible or audible artifacts when changing across different sonic characteristics. The proximity in the 1D space of the preamplifier or the tape recorder, the first creating harmonic distortion and the latter introducing time fluctuations, interferes more with learning the other effect. We showed two conditioning cases: the first uses a parameter to navigate a 1D space and interpolate among the two learned effects; the second uses three parameters to determine the amount of the individual effects to add to the hybrid mixture of effects. The design of the parameters and how to associate them with the different effects influence the user's morphing control. Different designs offer different interaction affordances. We demonstrate that specific neural network architecture, even with a small number of parameters, can learn different nonlinear sound alteration characteristics, such as those of distortion and compression, and temporal profiles of such effects, such as the time-variant characteristics of the compression and time shifting of the tape delay. Moreover, the approach can be expanded to other effects, but the selected model must be able to learn each effect separately in a black-box manner. In addition, selecting effects with similar sound altering characteristics can result in more accurate hybrid neural effects, such as modeling multiple distortion devices.

In future work, the interoperability of the models will be further investigated by including control parameters of individual audio effects as additional conditioning parame-

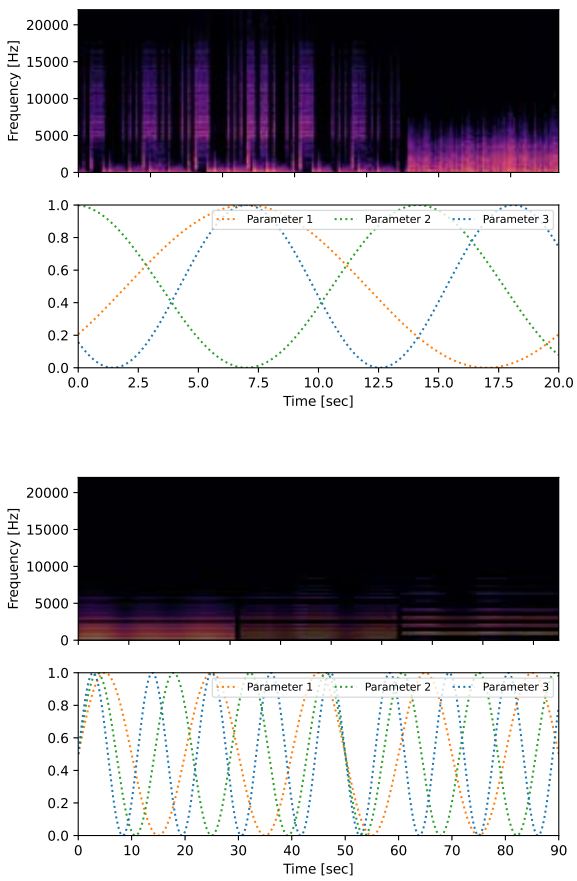


Figure 9. Normalized spectrogram of a 20 second music signal (top) and a 90 second sawtooth waves at 262.5, 525, and 1050 Hz (bottom), produced by the CTP model when the parameter is continuously changed by modulating it using a sine wave at 0.05, 0.07, and 0.09 Hz.

ters. This requires working with a homogeneous set of audio effects that present similar or compatible control parameters. Introducing user-control parameters to further condition the network can expand and enrich the sonic alteration capability of the hybrid audio effect.

6. REFERENCES

[1] D. S. Mendoza, *Emulating electric guitar effects with neural networks*, Barcelona, Spain, 2005.

[2] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep learning for tube amplifier emulation," in *Proc. of 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 2019.

[3] J. Covert and D. L. Livingston, "A vacuum-tube guitar amplifier model using a recurrent neural network," in *Proc. of 2013 IEEE Southeastcon*, Jacksonville, Florida, USA, 2013.

[4] E.-P. Damskägg, L. Juvela, V. Välimäki *et al.*, "Real-time modeling of audio distortion circuits with deep learning," in *Proc. of 16th Int. Sound and Music Computing Conf. (SMC)*, Malaga, Spain, 2019.

[5] J. Chowdhury, "A comparison of virtual analog modelling techniques for desktop and embedded implementations," *arXiv preprint arXiv:2009.02833*, 2020.

[6] C. J. Steinmetz and J. D. Reiss, "Efficient neural networks for real-time analog audio effect modeling," in *152nd Audio Engineering Society Convention*, The Hague, Netherlands, 2022.

[7] E. Perez, F. Strub, H. De Vries, V. Dumoulin, and A. Courville, "Film: Visual reasoning with a general conditioning layer," in *Proc. of 2018 Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, 2018.

[8] R. Simionato and S. Fasciani, "Deep learning conditioned modeling of optical compression," in *Proc. of 25th Int. Conf. Digital Audio Effects (DAFx)*, Vienna, Austria, 2022.

[9] ——, "Fully conditioned and low-latency black-box modeling of analog compression," in *Proc. of 26th Int. Conf. Digital Audio Effects (DAFx)*, Copenaghen, Denmark, 2023.

[10] A. Wright, V. Valimaki, O. Mikkonen, and E. Moliner, "Neural modeling of magnetic tape recorders," in *Proc. of 26th Int. Conf. Digital Audio Effects (DAFx)*, Copenaghen, Denmark, 2023.

[11] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," in *Proc. of Int. Conf. Learning Representations (ICLR-2020)*, Brighton, UK, 2020.

[12] A. Wright and V. Valimaki, "Neural modeling of phaser and flanging effects," *Journal of the Audio Engineering Society*, vol. 69, no. 7/8, pp. 517–529, 2021.

[13] M. Camras and D. W. Martin, "Magnetic recording handbook," 1989.

[14] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, "Language modeling with gated convolutional networks," in *International conference on machine learning*, Sydney, Australia, 2017.

[15] M. A. Martínez Ramírez, E. Benetos, and J. D. Reiss, "Deep learning for black-box modeling of audio effects," *Applied Sciences*, vol. 10, no. 2, p. 638, 2020.

[16] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. of 2015 Int. Conf. Learning Representations (ICLR)*, 2015.

[17] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *International conference on machine learning*, Atlanta, USA, 2013.

IMPLEMENTATION OF A CONTINUOUSLY VARIABLE DELAY LINE BY CROSSFADING BETWEEN SEVERAL TAP DELAYS

Thibaut CARPENTIER (thibaut.carpentier@ircam.fr) [1]

[1] **STMS Lab, IRCAM — CNRS — Sorbonne Université — Ministère de la Culture**, 1, place Igor-Stravinsky, Paris, 75004 France

ABSTRACT

Delay lines are ubiquitous in computer music applications: they are used to create audio effects or to simulate the sound propagation of moving sources. When one wants to synthesize a delay which varies over time, two main strategies are typically implemented: a) by using a "fractional" delay line, which simulates a non-integer delay based on an interpolator filter; or b) by applying a crossfade between the initial delay and the new desired delay value. These two techniques may induce pitch shift and/or spectral artifacts which are not tolerable in certain applications such as spatial reproduction by wavefield synthesis (WFS). In this article we propose a new method for creating a continuously variable delay line; the technique is an extension of the crossfade delay technique which exploits a superposition of several "auxiliary" tap delays whose times and gains are determined by a methodology similar to fractional delay interpolating filters. We show that it is thus possible to reduce coloration artifacts, at the expense of a higher computational cost.

1. INTRODUCTION

Delay lines (DL) are ubiquitous in signal processing and computer music applications. They are notably used for the production of audio effects such as vibrato, flanging, chorus, phasing [1] [2, Chapter 2] [3, Chapter 2] [4, Chapter 7.8], for physical modeling synthesis of musical instruments [5–7], for sound spatialization [8, 9] and artificial reverberation [10, 11], and for the simulation of propagation of moving sources, as observed in the Doppler effect or Leslie cabinet [12–14] or in auralization systems [15, Section 3.2] [16, Fig. 22] [17], etc. They also play a central role for sample rate conversion [18, 19], beamforming [20], and synchronization in wireless communication systems [21].

In many use cases, it is necessary to vary the delay value over time. When delay lines are implemented in the digital domain, the delay variation requires special attention. A "basic" delay line is generally implemented as a circular buffer, with two pointers which represent the read and write positions respectively, and which are incremented at each time step. In the digital domain, the read and write positions are necessarily integer values (expressed in number of samples). It is known that, for such a "basic" delay line, the transition from a delay τ_1 to a delay τ_2 (where τ_1 and τ_2 are rounded to the nearest integer values) induces a discontinuity in the output signal, which manifests itself as audible artifacts such as clicks or "zipper noise" [3, Chapter 2] [7].
To overcome this problem, it is necessary to implement an interpolation process in order to smooth the variations and minimize (or eliminate) the discontinuities. Two main strategies are listed in the literature, and very widely used in practice: a) implement a fractional delay line, using an interpolator filter to simulate non-integer delay values; b) perform a crossfade between two read pointers corresponding to the delays τ_1 and τ_2. These approaches each have limitations — which will be discussed in more detail in the following sections: a) a fractional delay line induces a pitch shift comparable to a Doppler effect; depending on the use cases, this transposition can be artistically relevant or undesirable; b) the crossfade technique generates a comb filtering effect with detuning (undesired distorsion of the magnitude and phase spectrum) which can be more or less pronounced depending on the situation.

Spatial reproduction of sound sources by wavefield synthesis (WFS) [9, 22] is a use case of interest, and the initial motivation of this study. In WFS, each loudspeaker (called "secondary source") of the reproduction system contributes to recreating the wave front emanating from a virtual source (or "primary source"), and the driving functions of the loudspeakers — derived from the Kirchhoff-Helmholtz and Rayleigh I integral equations — involve delay lines (a priori with fractional delay values). Each time the virtual source moves, the relative delays of all the speakers should vary continuously. Given that the number of secondary sources is generally large (several dozens to hundreds) and that the range of the delays can be considerable (up to several tens of milliseconds depending on the configuration and source movements), WFS reproduction is particularly sensitive to the delay line artifacts discussed previously. The pitch changes of a fractional delay and the spectral colorations of a crossfade delay are both undesirable [19, 23–26].
In this article we propose a new method for creating a continuously variable delay line; the technique is an extension of the crossfade delay that exploits "auxiliary" tap delays whose times and gains are determined by a methodology similar to fractional delay interpolating filters. We will show that it is therefore possible to reduce coloration

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

artifacts, at the expense of a higher computational cost.

The article is organized as follows: first, we present elementary reminders on fractional delay lines (Section 2), and crossfading delay lines (Section 3). In Section 4, we present and evaluate the proposed method, before concluding.

2. FRACTIONAL DELAY LINE (FDL)

2.1 Ideal response

Fractional delay lines (FDL) allow to simulate any delay $\tau \in \mathbb{R}^+$. The delay value is decomposed into an integer part and a fractional part: $\tau = \tau_i + \beta$ where $\tau_i = \lfloor \tau \rfloor$ is the integer part and $\beta \in [0, 1) = \tau - \tau_i$ is the fractional part. For an input sequence $x[n]$ originating from the discretization of a band-limited analog signal $x(t)$, the output of a fractional delay line is written $y[n] = x[n - (\tau_i + \beta)]$. This value a priori lies "between two samples", which is not possible. Instead, one must determine the discrete values (on the sampling grid) by interpolation.
Many methods have been proposed and extensively discussed (see in particular [1, 5, 27–29] [4, Chapter 5.5]). These techniques generally rely on the approximation of an "ideal delay operator" by a finite impulse response filter (FIR) or infinite impulse response filter (IIR). Considering the relation $Y(z) = z^{-\tau} X(z)$, the ideal transfer function is written in the frequency domain:

$$H_{\text{ideal}}\left(e^{i\omega}\right) = e^{-i\omega\tau} \ , \tag{1}$$

where $z = e^{i\omega}$, and $\omega = 2\pi\, f/f_s$ denotes the angular frequency. As a result, the impulse response of the system writes $\forall n$

$$h_{\text{ideal}}[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_{\text{ideal}}\left(e^{i\omega}\right)\, e^{i\omega n}\, d\omega = \text{sinc}\,(n - \tau) \tag{2}$$

where $\text{sinc}\,(x) = \frac{\sin(\pi x)}{\pi x}$ is the cardinal sine function.
When the desired delay τ is integer, the response reduces to a single pulse at the discrete time $n = \tau$, as in "basic" DL. When τ is real and non-integer, the impulse response $h_{\text{ideal}}[n]$ has an infinite length, and is non-causal. It is therefore not suitable for real-time applications. The FDL must be implemented with an approximate solution.

2.2 Approximations of the ideal solution

A large number of approximation techniques have been proposed in the literature. The simplest and most intuitive approach relies on approaching the ideal delay operator by linear interpolation between adjacent samples $x[n-1]$ and $x[n]$. This technique has a low computational cost, and it provides qualitatively satisfactory results when the spectrum of the signal x is mainly concentrated in low frequencies. To process large-band signals, it is preferable to apply oversampling, however this considerably increases the computational cost.

Another simple way to approximate the ideal operator h_{ideal} is to truncate the reponse, and shift it temporally to ensure causality. This results in an FIR filter of order M. It is known that the impulse response of this filter is subject to the Gibbs phenomenon, and consequently its magnitude spectrum exhibits ripples (see for example [4, Figure 5.46]), which is generally undesirable.

Another widely used approach consists of implementing the FDL using a 1st order all-pass IIR filter [27] [4, Chapter 5.5.3], known as 1st order Thiran filter [30] [29, Section 20.4.2.2.], which guarantees a maximally flat group delay for $\omega \to 0$. The approach has several advantages: the solution is obtained analytically, the computation cost is very low, and the frequency response is flat for all frequencies. However, the phase delay approximation is only valid in a domain restricted to low frequencies, and degrades rapidly at higher frequencies. Furthermore, the impulse response is relatively long, which is problematic for transient sounds, when the input signal or the all-pass coefficient varies quickly (causing audible clicks) [5, 31]. Alternative implementations have been proposed to mitigate this problem [25,32,33], but at the expense of higher complexity.

Finally, another commonly used strategy approximates the desired function by an interpolator polynomial of order N [27] [4, Chapter 5.5.4]. The simplest analytical solution is provided by the Lagrange polynomial. The coefficients of the FIR Lagrange interpolator can be found e.g. in [27] [29, Section 20.3.4.]. For $N = 1$, it is equivalent to linear interpolation; for $N \to \infty$, it converges to the ideal sinc operator [7]. The filter exhibits a low-pass behavior, more pronounced for even orders N. As a consequence, order $N = 3$ is frequently used, as it offers a tradeoff between moderate computational cost, and acceptable magnitude/phase response — the frequency response is almost flat up to $f_s/4$, see Figure 2(a). Figure 1 displays an example impulse response of the Lagrange interpolator filter of order $N = 3$.

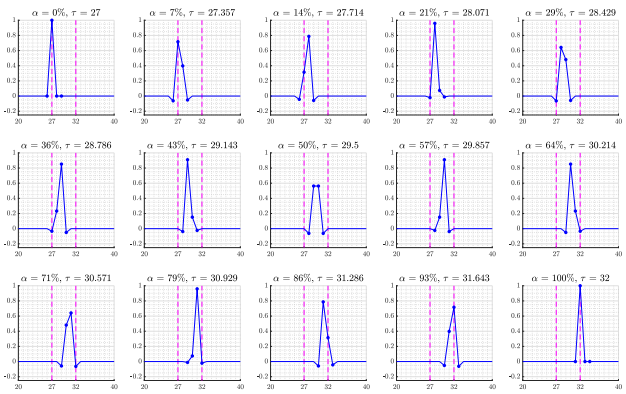


Figure 1. Impulse response of a FDL for $\tau_1 = 27$, $\tau_2 = 32$, and for different values of the interpolation factor α between 0% and 100%. The x-axis and y-axis represent time (in samples) and amplitude respectively. The purple dashed lines symbolize τ_1 and τ_2. The FDL is implemented with 4-point Lagrange interpolation ($N = 3$).

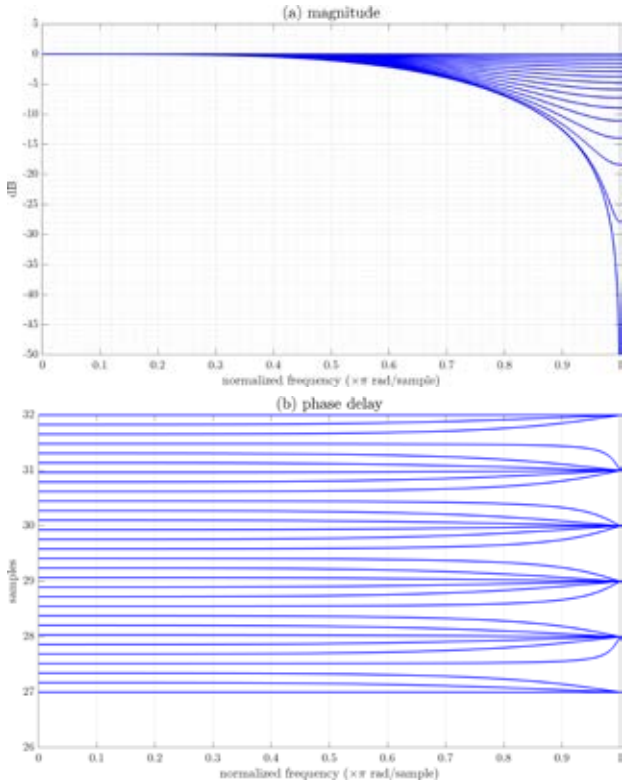


Figure 2. (a) Frequency response and (b) phase delay of a FDL with Lagrange interpolation. The simulated parameters are similar to Figure 1. The frequency response is most attenuated when $\alpha = 50\%$, with $\left\| H\left(e^{i\omega}\right)\big|_{\omega=\pi} \right\| = 0$.

2.3 Artifacts when varying the delay value

The accuracy of the sinc ideal response approximation is generally improved by using higher order interpolation schemes, however the computational cost then becomes prohibitive for real-time applications. Furthermore, whatever the method used to implement the FDL, audible artifacts are generated when the delay value varies continuously over time such that $y(t) = x\big(t - \tau(t)\big)$. For the sake of simplicity, we assume for example that $\tau(t)$ varies linearly between two integer delays τ_1 and τ_2. Without loss of generality, we will also assume that $\tau_2 > \tau_1$. We note α the "interpolation" factor such that, at each time t, the (fractional) delay τ is worth

$$\tau(t) = \alpha\,\tau_1 + (1-\alpha)\,\tau_2\ , \tag{3}$$

with $\alpha \in \mathbb{R}$ and $0 \leq \alpha \leq 1$. Let $\Delta = \tau_2 - \tau_1$ be the delay variation, and δ be the interpolation duration to go from τ_1 to τ_2, i.e. $\alpha\big|_{t=0} = 1$ and $\alpha\big|_{t=\delta} = 0$.

By varying τ, a momentary transposition is generated; this is analogous to the Doppler effect and results in a clearly audible pitch change [1, 7] [34, Chapter 7.7] [35]. This Doppler effect is physically correct: the transition from τ_1 to τ_2 amounts to changing the physical distance which separates a listener from a sound source (the delay line realistically "modeling" the acoustic propagation channel). When $\Delta > 0$, the transposition is downward; and vice versa when $\Delta < 0$ [34, 36]. As noted by Puckette [34, Chapters 7.7 and 7.8], this effect is quite pronounced even when the interpolation duration (δ) is significantly greater than the delay variation (Δ); the effect of pitch shift is for example clearly audible even with long duration $\delta > 100\,|\Delta|$ (By comparison, when we consider amplitude variations, with gain ramps, satisfactory results can generally be obtained even with relatively short interpolation times, e.g. $\delta \approx 20$ ms for $|\Delta| \approx 100$ dB).

In some applications, the induced Doppler effect may be acceptable or even desirable. In fact, this justifies the ubiquitous use of delay lines in audio effects implementing pitch variations or modulations (vibrato effect, for example) [34, Chapter 7.9] [29, Section 20.7.4.] [1, 37].
On the other hand, in spatialization or auralization contexts, the Doppler shift – even though it simulates a realistic auditory effect – is sometimes undesirable, considered disruptive and harmful to timbral content [19, 23, 38]. As mentioned in the introduction, this is all the more critical for WFS rendering, if the virtual sources move "rapidly" (generating variations $\Delta \gg 1$, of the order of a few tens of milliseconds or more), and if the number of loudspeakers is high.
Similarly, in auralization systems, delay lines are used to simulate the acoustic channel (i.e. the reflection paths) between a sound source and a listener; when one or the other moves, the delays must be continually adjusted. Depending on the simulated geometric configuration, the number of simulated delays (acoustic reflection paths) can be significant; the constraints are therefore similar to the WFS rendering presented above.

3. CROSSFADE BETWEEN 2 TAP DELAYS (XDL)

3.1 Principle

When the Doppler pitch shift is undesirable, or when $\Delta \gg 1$, the alternative method of crossfading delay line (XDL) may be used. This approach consists of using a delay line with two tap delays (i.e. two read pointers), corresponding to τ_1 and τ_2 respectively. To simulate a variation of $\tau(t)$ over time, a crossfade is applied i.e. the amplitude of the two tap delays is interpolated for a duration δ. For the sake of simplicity, a linear crossfade is commonly used, but other types of transition curves can also be implemented.
The XDL technique is rarely discussed in the literature, probably because it is conceptually very simple, and not relying on any physical foundation. Nevertheless, it is briefly mentioned in several publications and patents, for instance concerning the creation of legato effect in the waveguide synthesis of string instruments [39–42]. It is also mentioned in [7, Paragraph "Large Delay Changes"] [43].
Despite its low representation in the academic literature, the XDL approach is very commonly used in computer music tools. For example, it is implemented under the name "Double Delay with Interpolation (DDI)" or "sdelay" (*smooth delay*) in the `Faust` libraries [44]. The crossfade of tap delays is also used (under the name "xdelay") in the WFS implementation of `Spat` [45] and in the mixing

workstation `Panoramix` [46]. In the Max environment, several implementations can be found such as `vdb~` (by Benjamin Thigpen), `ej.vdb~` (by Emmanuel Jourdan), or `M4L.vdelay~` (included in `Max for Live`).

3.2 Implementation

The software implementation of the XDL method is straightforward, and does not raise any particular challenges. However, several remarks can be made — see also [7]. a) A single delay line is sufficient, with one write pointer and two read pointers; thus the memory space requirement is comparable to an FDL. b) During the crossfade period, the computation cost is doubled. c) The duration δ of the crossfade should be long enough to achieve smooth results. However, there is no general rule for choosing δ as a function of Δ, the quality of the rendering may depend on the nature of the input signal. d) The XDL approach is compatible with non-integer τ_1 and τ_2 delays; in this case, the read pointers will use one of the fractional interpolation techniques presented in Section 2. To guarantee an artifact-free result, it may then be wise to take into account the length of the transient response of the interpolator filter: for example, when using an order N filter, one can "pre-warm" the filter for $N+1$ samples before starting the crossfade operation. However, if the duration of the fade δ is large compared to the duration of the transient response of the interpolator filter, the "pre-warming" step is not essential (the transient response of the filter is not audible).

3.3 Limitations

Figure 3 shows the impulse response of an XDL, for different values of the interpolation factor α between 0% and 100%. While the system performs a kind of morphing between the two pure delays τ_1 and τ_2, it does not achieved any interpolation, i.e. it does not synthesize any actual intermediate delay values.

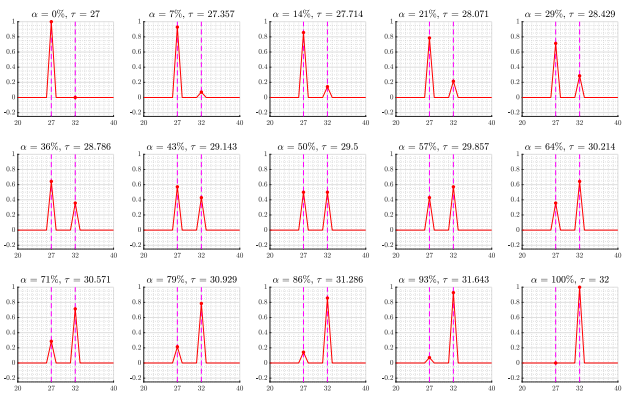


Figure 3. Impulse response of an XDL, for $\tau_1 = 27$, $\tau_2 = 32$, and for various values of the interpolation factor α between 0% and 100%. The x-axis and y-axis represent time (in samples) and amplitude respectively. The vertical purple dashed lines symbolize τ_1 and τ_2.

The transfer function of the system is

$$\begin{aligned} H(z) &= \alpha\, z^{-\tau_1} + (1-\alpha)\, z^{-\tau_2} \\ &= \alpha\, z^{-\tau_1} \left(\frac{z^{\Delta} + \frac{1-\alpha}{\alpha}}{z^{\Delta}} \right) . \end{aligned} \tag{4}$$

In the righthand term we recognize the typical response of a feedforward comb filter [4, Chapter 5.2]. The frequency response is shown in Figure 4(a); it exhibits notches for the pulsations $\omega = \frac{\pi}{\Delta}, \frac{3\pi}{\Delta}, \frac{5\pi}{\Delta}, \ldots$ The (linear) amplitude of these notches is zero when $\alpha = 50\%$. The phase delay (Figure 4(b)) also exhibits some distortions.
In fact, XDL crossfading is subject to the typical comb filtering coloration. The perception of this spectral coloration may be affected by several factors (input signal, Δ, δ). Depending on the target applications, the artifacts can be tolerated (or sometimes inaudible), or they may be deemed unacceptable.

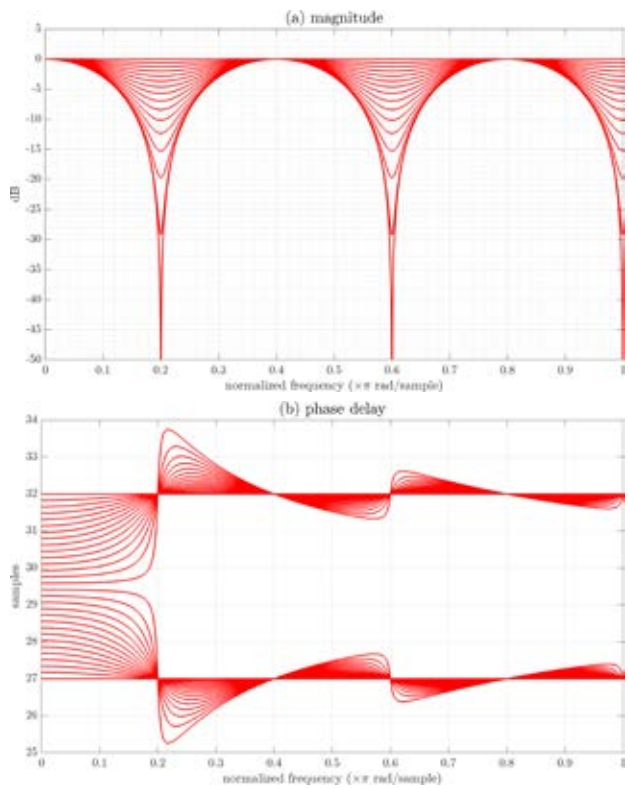


Figure 4. (a) Frequency response and (b) phase delay of an XDL for $\tau_1 = 27$, $\tau_2 = 32$, and for different values of the interpolation factor α between 0% and 100%. The frequency response shows periodic notches for angular frequencies $\frac{\omega}{\pi} = \frac{1}{\Delta} = 0.2$, $\frac{\omega}{\pi} = \frac{3}{\Delta} = 0.6$, and $\frac{\omega}{\pi} = \frac{5}{\Delta} = 1$.

4. PROPOSED METHOD: CROSSFADE BETWEEN $(2K+2)$ TAP DELAYS

4.1 Principle

In order to overcome the limitations of the two previous approaches, we propose a new method which is somehow a compromise between FDL and XDL. We take the XDL

as a starting point; instead of reading and fading between two tap delays τ_1 and τ_2, we propose to add other auxiliary tap delays in order to "temper" the excessively strong comb filtering effect. It is then a matter of choosing the discrete times τ_k and the amplitudes of these auxiliary delays over time. For simplicity, we assume that the start and end delays τ_1 and τ_2 are integers. It seems obvious that the auxiliary delays must be symmetric with respect to τ_1 and τ_2, as there is no reason to favor one or the other. To limit the computational cost, it is desirable that the auxiliary delays t_k be integer values (but this is not a necessary condition). Finally, for the sake of simplicity, it is preferable that the delays t_k are regularly sampled (it is not proven that this criterion is necessary). Therefore, we suggest using a "sampling step" Δ, so that the delay values be $\{\tau_1 - k\Delta\}$ and $\{\tau_2 + k\Delta\}$, with $k = 0, 1, ..., K$. In other words, the delays t_k write

$$\begin{cases} t_k = \tau_1 - (K-k)\,\Delta\,, \text{ if } 0 \leq k \leq K \\ t_k = \tau_2 + (k-K-1)\,\Delta\,, \text{ if } (K+1) \leq k < (2K+2)\,. \end{cases} \tag{5}$$

This results in $(2K+2)$ tap delays in total. Note that the XDL presented in Section 3 is a particular case corresponding to $K = 0$.

We then need to determine the crossfading law, that is to say, for each time t, the amplitude associated with each delay t_k. By analogy with the reasoning of Section 2.1, we ideally seek a solution in the form of a sinc function, reaching its maximum at $t = \tau$, however with a "width" Δ:

$$h_{\text{ideal}}(t) = \text{sinc}\left(\frac{t-\tau}{\Delta}\right)\,. \tag{6}$$

In the discrete-time domain, we can therefore approximate the ideal response by truncation:

$$0 \leq k < (2K+2), \quad h[k] = \text{sinc}\left(\frac{t_k - \tau}{\Delta}\right)\,. \tag{7}$$

This principle is illustrated in Figure 5, for $K = 4$.

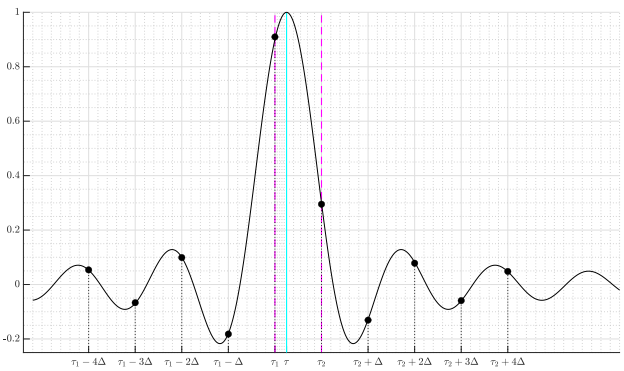


Figure 5. Principle of the proposed method: approximation of the ideal impulse response by truncation. Example with 10 tap delays (i.e. $K = 4$). The vertical purple dashed lines symbolize τ_1 and τ_2. The cyan line symbolizes $\tau\,(t)$, here for $\alpha = 25\%$. The solid black line depicts $\text{sinc}\left(\frac{t-\tau}{\Delta}\right)$. The black markers represent $h[k]$ for $0 \leq k < (2K+2)$.

For causality reasons, it is required that $\tau_1 - k\Delta \geq 0$, which puts a constraint on K such that $K \leq \left\lfloor \frac{\tau_1}{\Delta} \right\rfloor$.

Furthermore, if we note L the allocated length of the delay line, the constraint $\tau_2 + k\Delta \leq L$ must be verified, hence imposing: $K \leq \left\lfloor \frac{(L-\tau_2)}{\Delta} \right\rfloor$. This leads to:

$$K \leq \min\left(\left\lfloor \frac{\tau_1}{\Delta} \right\rfloor, \left\lfloor \frac{(L-\tau_2)}{\Delta} \right\rfloor\right)\,. \tag{8}$$

4.2 Results

The impulse response of the proposed system is displayed in Figure 6 for different values of the interpolation factor α ranging from 0% to 100%. The behavior is analogous to Figure 3, however with auxiliary "pulses", some of which exhibit a negative amplitude. Note that the amplitude of the two main delays (for $t = \tau_1$ and $t = \tau_2$) is different from the Figure 3.

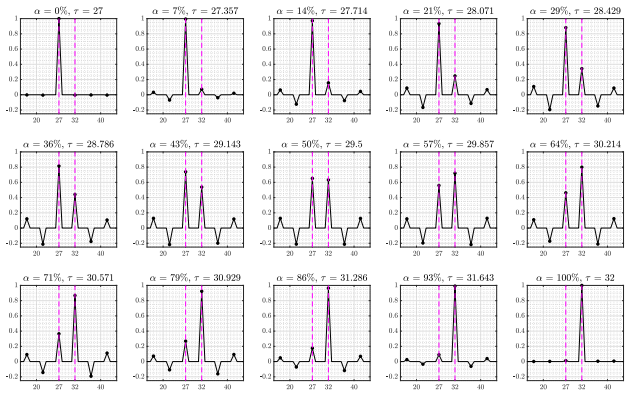


Figure 6. Impulse response of the proposed method (here with $K = 2$), for $\tau_1 = 27$, $\tau_2 = 32$, and for different values of the interpolation factor α between 0% and 100%. The x-axis and y-axis represent time (in samples) and amplitude respectively. The purple dashed lines symbolize τ_1 and τ_2.

Figure 7 illustrates the frequency response of the system, for different values of the interpolation factor α between 0% and 100%. The phase delay generally behaves like the XDL (Figure 4(b)). The magnitude spectrum exhibits notches for the same angular frequencies $\omega = \frac{\pi}{\Delta}, \frac{3\pi}{\Delta}, \frac{5\pi}{\Delta}, ...$, but these notches are significantly narrower than for XDL (Figure 4(a)). In other words, the undesirable comb filtering effect is reduced. Apart from these notches, we also observe slight ripples around 0 dB, which are undesired; however the range of these ripples remains moderate (less than 2 dB in this example).

4.3 Influence of the number of tap delays

It is obvious that the number K plays a preponderant role. By increasing K (within the authorized range Eq. 8), the spectral effect of comb filtering is reduced. To visualize this effect, we superimpose in Figure 9 the frequency curves for different values of K, and in the worst-case scenario $\alpha = 50\%$. We observe that an increase of K significantly reduces the width of the notches.

Figure 8 presents similar results, now in the case $\alpha = 25\%$;

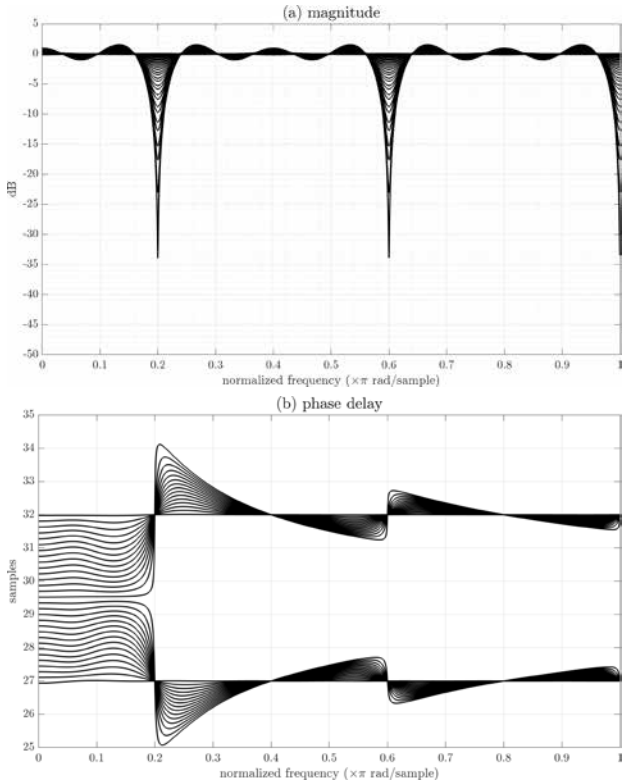


Figure 7. (a) Frequency response and (b) phase delay for the proposed method for $K = 2$, $\tau_1 = 27$, $\tau_2 = 32$, and for different values of the interpolation factor α between 0% and 100%.

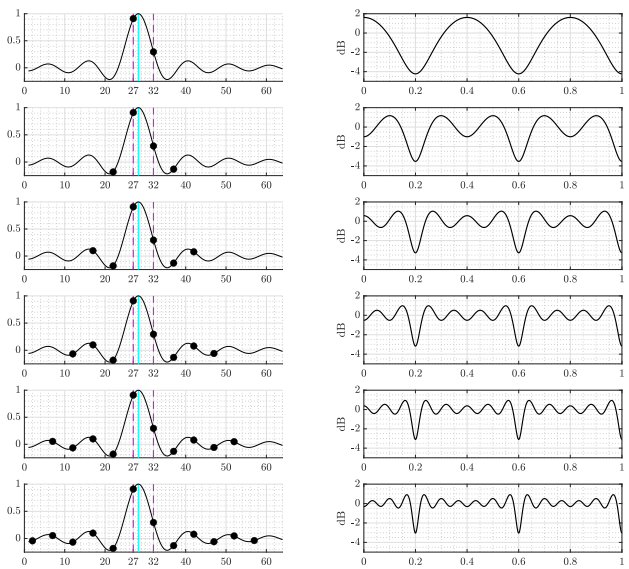


Figure 8. The curves are for $\tau_1 = 27$, $\tau_2 = 32$, and $\alpha = 25\%$ (i.e. $\tau = 28.25$ samples). Left column: impulse response for the proposed method. In solid black line, the function $\mathrm{sinc}\left(\frac{t-\tau}{\Delta}\right)$. The cyan vertical line symbolizes $\tau = 28.25$. Right column: frequency response. First line: $K = 0$; Second line: $K = 1$; Third line: $K = 2$; etc.

it also shows the impulse response for $K = 0, 1, ..., 5$.

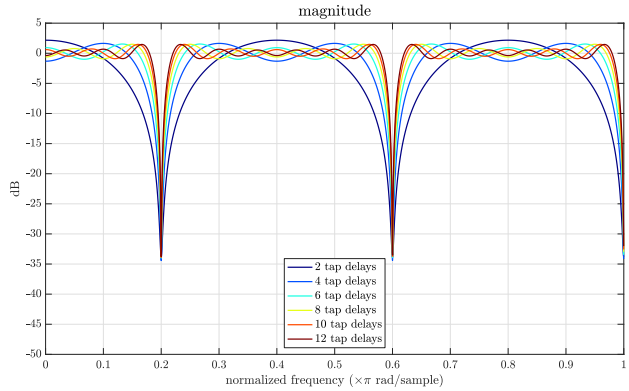


Figure 9. Frequency response of the proposed method, for $\tau_1 = 27$, $\tau_2 = 32$, $\alpha = 50\%$, and $K = 0, 1, 2, ..., 5$.

The computational cost of the proposed method is obviously proportional to the total number of delays $(2K + 2)$. The choice of K is necessarily a compromise. For WFS applications, where the number of delay lines may be high (depending on the number of primary and secondary sources), it is preferable to restrict to $K = 1$, which already offers an audible improvement compared to XDL (based on informal listening).

5. CONCLUSION

In this article we proposed a new method for creating a continuously variable delay line. When the delay varies over time, we operate a crossfade between several "auxiliary" read pointers. The times and gains of these auxiliary tap delays are chosen so as to minimize the typical comb filtering coloration issue of XDL. The introduction of auxiliary tap delays, however, has a significant impact on the computational cost. In future work, a more precise evaluation of the method's performance remains to be carried out, in particular in comparison with alternative approaches such as "samplerate conversion" [18, 19, 24, 47, 48], or Lagrange interpolator filter implemented in the form of a Farrow structure [21, 49].
Likewise, it would be necessary to conduct a perceptual study of residual artifacts.

6. REFERENCES

[1] J. Dattorro, "Effect Design : Part 2: Delay-Line Modulation and Chorus," *Journal of the Audio Engineering Society*, vol. 45, no. 10, pp. 764 – 788, October 1997. https://www.aes.org/e-lib/browse.cfm?elib=10159

[2] U. Zölzer, *DAFX: Digital Audio Effects*, 2nd ed. John Wiley & Sons, March 2011. https://doi.org/10.1002/9781119991298

[3] J. D. Reiss and A. P. McPherson, *Audio Effects: Theory, Implementation and Application*, 1st ed. Boca Raton, FL, USA: CRC Press, January 2015. https://doi.org/10.1201/b17593

[4] A. Uncini, *Digital Audio Processing Fundamentals*, 1st ed. Cham, Switzerland: Springer International Publishing, February 2022. https://doi.org/10.1007/978-3-031-14228-4

[5] V. Välimäki, "Discrete-Time Modeling of Acoustic Tubes Using Fractional Delay Filters," Ph.D. dissertation, Helsinki University of Technology, Helsinki, Finland, December 1995.

[6] S. Bilbao, *Numerical Sound Synthesis : Finite Difference Schemes and Simulation in Musical Acoustics*. John Wiley & Sons, October 2009. https://doi.org/10.1002/9780470749012

[7] J. O. Smith, *Physical Audio Signal Processing: for Virtual Musical Instruments and Digital Audio Effects*. W3K Publishing, December 2010. https://ccrma.stanford.edu/~jos/pasp/

[8] F. Rumsey, *Spatial Audio*, 1st ed. New York, NY, USA: Routledge, June 2001. https://doi.org/10.4324/9780080498195

[9] S. Spors, H. Wierstorf, A. Raake, F. Melchior, M. Frank, and F. Zotter, "Spatial Sound With Loudspeakers and Its Perception: A Review of the Current State," *Proc. of the IEEE*, vol. 101, no. 9, pp. 1920 – 1938, September 2013. https://doi.org/10.1109/jproc.2013.2264784

[10] W. G. Gardner, "Reverberation Algorithms," in *Applications Of Digital Signal Processing To Audio And Acoustics*, M. Kahrs and K. Brandenburg, Eds. New York, NY, USA: Springer, 2002, ch. 3, pp. 85 – 131. https://doi.org/10.1007/0-306-47042-X_3

[11] V. Välimäki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty Years of Artificial Reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1421 – 1448, July 2012. https://doi.org/10.1109/tasl.2012.2189567

[12] J. O. Smith, S. Serafin, J. S. Abel, and D. Berners, "Doppler Simulation and the Leslie," in *Proc. of the 5th International Conference on Digital Audio Effects (DAFx)*, Hamburg, Germany, September 2002.

[13] R. Kronland-Martinet and T. Voinier, "Real-Time Perceptual Simulation of Moving Sources: Application to the Leslie Cabinet and 3D Sound Immersion," *EURASIP Journal on Audio, Speech, and Music Processing*, pp. 1 – 10, 2008. https://doi.org/10.1155/2008/849696

[14] Z.-H. Fu and T. Zhu, "Moving Sound Image Rendering in Realistic Audio Effects," in *Intelligent Science and Intelligent Data Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 722 – 730. https://doi.org/10.1007/978-3-642-31919-8_92

[15] T. Lokki, "Physically-based Auralization - Design, Implementation, and Evaluation," Ph.D. dissertation, Helsinki University of Technology, Espoo, Finland, November 2002.

[16] L. Savioja, J. Huopaniemi, T. Lokki, and R. Väänänen, "Creating Interactive Virtual Acoustic Environments," *Journal of the Audio Engineering Society*, vol. 47, no. 9, pp. 675 – 705, September 1999. http://www.aes.org/e-lib/browse.cfm?elib=12095

[17] P. Schäfer, J. Fatela, and M. Vorländer, "Interpolation of scheduled simulation results for real-time auralization of moving sources," *Acta Acustica*, vol. 8, p. 9, 2024. https://doi.org/10.1051/aacus/2023070

[18] A. Tarczynski, W. Kozinski, and G. Cain, "Sampling rate conversion using fractional-sample delay," in *Proc. of IEEE ICASSP*, Adelaide, SA, Australia, April 1994. https://doi.org/10.1109/icassp.1994.390042

[19] A. Franck, "Efficient Algorithms for Arbitrary Sample Rate Conversion with Application to Wave Field Synthesis," Ph.D. dissertation, Universitätsverlag Ilmenau, Ilmenau, Germany, 2012.

[20] P. Murphy, A. Krukowski, and A. Tarczynski, "An efficient fractional sample delayer for digital beam steering," in *Proc. of the IEEE ICASSP*, Munich, Germany, April 1997, pp. 2245 – 2248. https://doi.org/10.1109/icassp.1997.599498

[21] C. Farrow, "A continuously variable digital delay element," in *Proc. of the IEEE International Symposium on Circuits and Systems*, Espoo, Finland, June 1988. https://doi.org/10.1109/iscas.1988.15483

[22] R. Rabenstein, S. Spors, and P. Steffen, "Wave Field Synthesis Techniques for Spatial Sound Reproduction," in *Topics in Acoustic Echo and Noise Control*, E. Hänsler and G. Schmidt, Eds. Springer, 2006, ch. 13, pp. 517 – 545. https://doi.org/10.1007/3-540-33213-8_13

[23] A. Franck, A. Graefe, T. Korn, and M. Strauss, "Reproduction of Moving Sound Sources by Wave Field Synthesis: An Analysis of Artifacts," in *Proc. of the AES 32nd International Conference: DSP For Loudspeakers*, Hillerød, Denmark, September 2007. https://www.aes.org/e-lib/browse.cfm?elib=14206

[24] A. Franck, K. Brandenburg, and U. Richter, "Efficient Delay Interpolation for Wave Field Synthesis," in *Proc. of the 125th Convention of the Audio Engineering Society (AES)*, San Francisco, CA, USA, October 2008. https://www.aes.org/e-lib/browse.cfm?elib=14764

[25] F. Winter and S. Spors, "On fractional delay interpolation for local wave field synthesis," in *Proc. of the 24th European Signal Processing Conference (EUSIPCO)*, Budapest, Hungary, August 2016. https://doi.org/10.1109/eusipco.2016.7760682

[26] J. Ahrens and S. Spors, "Reproduction of Moving Virtual Sound Sources with Special Attention to the Doppler Effect," in *Proc. of the 124th Convention of the Audio Engineering Society (AES)*, Amsterdam, Netherlands, May 2008. http://www.aes.org/e-lib/browse.cfm?elib=14493

[27] T. I. Laakso, V. Välimäki, M. Karjalainen, and U. K. Laine, "Splitting the Unit Delay," *IEEE Signal Processing Magazine*, vol. 13, no. 1, pp. 30 – 60, January 1996. https://doi.org/10.1109/79.482137

[28] V. Välimäki and T. I. Laakso, "Principles of fractional delay filters," in *Proc. of the IEEE ICASSP*, Istanbul, Turkey, June 2000. https://doi.org/10.1109/ICASSP.2000.860248

[29] ——, "Fractional Delay Filters – Design and Applications," in *Nonuniform Sampling*, 1st ed., F. Marvasti, Ed. New York, NY, USA: Springer New York, November 2001, ch. 20, pp. 835 – 895. https://doi.org/10.1007/978-1-4615-1229-5_20

[30] J.-P. Thiran, "Recursive digital filters with maximally flat group delay," *IEEE Transactions on Circuit Theory*, vol. 18, no. 6, pp. 659 – 664, 1971. https://doi.org/10.1109/tct.1971.1083363

[31] S. A. Van Duyne, D. A. Jaffe, G. P. Scandalis, and T. S. Stilson, "A Lossless, Click-free, Pitchbendable Delay Line Loop Interpolation Scheme," in *Proc. of the International Computer Music Conference (ICMC)*, Thessaloniki, Greece, September 1997. http://hdl.handle.net/2027/spo.bbp2372.1997.068

[32] V. Välimäki, T. I. Laakso, and J. MacKenzie, "Elimination of Transients in Time-Varying Allpass Fractional Delay Filters with Applications to Digital Waveguide Modeling," in *Proc. of the International Computer Music Conference (ICMC)*, Banff, Canada, September 1995. http://hdl.handle.net/2027/spo.bbp2372.1995.096

[33] V. Valimaki and T. Laakso, "Suppression of transients in variable recursive digital filters with a novel and efficient cancellation method," *IEEE Transactions on Signal Processing*, vol. 46, no. 12, pp. 3408 – 3414, 1998. https://doi.org/10.1109/78.735314

[34] M. S. Puckette, *The Theory and Technique of Electronic Music*. World Scientific Press, May 2007. https://doi.org/10.1142/6277

[35] D. Griesinger, "Improving Room Acoustics Through Time-Variant Synthetic Reverberation," in *Proc. of the 90th Convention of the Audio Engineering Society (AES)*, Paris, France, February 1991. http://www.aes.org/e-lib/browse.cfm?elib=5679

[36] J. Pakarinen, M. Karjalainen, V. Valimaki, and S. Bilbao, "Energy behavior in time-varying fractional delay filters for physical modeling synthesis of musical instruments," in *Proc. of the IEEE ICASSP*, vol. 3, Philadelphia, PA, USA, March 2005. https://doi.org/10.1109/ICASSP.2005.1415631

[37] V. Lazzarini, J. Timoney, and T. Lysaght, "Adaptive FM Synthesis," in *Proc. of the 10th International Conference on Digital Audio Effects (DAFx)*, Bordeaux, France, September 2007.

[38] J. Jaworski and W. L. Martens, "A Novel Method for Simulating Movement in Multichannel Reverberation Providing Enhanced Listener Envelopment," in *Proc. of the AES International Conference on Spatial Reproduction - Aesthetics and Science*, Tokyo, Japan, August 2018. http://www.aes.org/e-lib/browse.cfm?elib=19660

[39] D. A. Jaffe and J. O. Smith, "Performance Expression in Commuted Waveguide Synthesis of Bowed Strings," in *Proc. of the International Computer Music Conference (ICMC)*, Banff, Canada, September 1995, pp. 343 – 346. http://hdl.handle.net/2027/spo.bbp2372.1995.098

[40] ——, "Patent WO1998010355A1 – Digital signal processing system and method for generating musical legato using multitap delay line with crossfader," Palo Alto, CA, USA, September 1997.

[41] J. O. Smith, "Physical Modeling Synthesis Update," *Computer Music Journal*, vol. 20, no. 2, p. 44, 1996. https://doi.org/10.2307/3681331

[42] S. A. V. Duyne, D. A. Jaffe, G. P. Scandalis, , and T. S. Stilson, "Patent 5 742 532 – System and method for generating fractional length delay lines in a digital signal processing system," April 1998.

[43] K. Bogdanowicz and R. Belcher, "Using Multiple Processors for Real-Time Audio Effects," in *Proc. of the AES 7th International Conference: Audio in Digital Times*, Toronto, ON, Canada, May 1989, pp. 337 – 342. http://www.aes.org/e-lib/browse.cfm?elib=5449

[44] J. O. Smith, "Signal Processing Libraries for Faust," in *Proc. of the Linux Audio Conference (LAC)*, Stanford, CA, USA, April 2012.

[45] T. Carpentier, "Spat: a comprehensive toolbox for sound spatialization in Max," *Ideas Sónicas*, vol. 13, no. 24, pp. 12 – 23, June 2021. https://hal.science/hal-03356292

[46] ——, "A versatile workstation for the diffusion, mixing, and post-production of spatial audio," in *Linux Audio Conference*, Saint-Etienne, France, May 2017. https://hal.archives-ouvertes.fr/hal-01527754

[47] A. Franck and K. Brandenburg, "An overall optimization method for arbitrary sample rate converters based on integer rate SRC and lagrange interpolation," in *Proc. of the IEEE WASPAA*, New Paltz, NY, USA, October 2009. https://doi.org/10.1109/aspaa.2009.5346472

[48] D. Rocchesso, "Fractionally addressed delay lines," *IEEE Transactions on Speech and Audio Processing*, vol. 8, no. 6, pp. 717 – 727, 2000. https://doi.org/10.1109/89.876310

[49] A. Franck, "Efficient Algorithms and Structures for Fractional Delay Filtering Based on Lagrange Interpolation," *Journal of the Audio Engineering Society*, vol. 56, no. 12, pp. 1036 – 1056, December 2008. https://www.aes.org/e-lib/browse.cfm?elib=14647

REAL-TIME PSYCHOACOUSTIC FREQUENCY MASKING COMPENSATION FOR AUDIO SIGNALS WITH OVERLAPPING SPECTRA

Giorgio PRESTI (giorgio.presti@unimi.it) (0000-0001-7643-9915) [1],
Nicola DEGIORGI (nicoladegiorg@gmail.com) [2], **Amedeo FRESIA** (amedeofresia@gmail.com) [2], and
Antonio SERVETTI (antonio.servetti@polito.it) (0000-0002-0159-5718) [2]

[1] *Laboratory of Music Informatics (LIM), Department of Computer Science*, **University of Milan**, Italy
[2] *Control and Computer Engineering Department*, **Politecnico di Torino**, Italy

ABSTRACT

This work presents a prototype for the automatic and real-time psychoacoustic frequency masking effect compensation. This tool is intended to handle audio signals with overlapping spectra coming from two different mono or stereo sources in order to emphasise those frequencies of a main source that the second one masks. The goal is achieved using a dynamic equaliser controlled by a function of the differences of the input spectra psychoacoustic models. The tool has been prototyped as a standard Virtual Studio Technology (VST3) plugin and its effectiveness has been tested with a user study carried out in the context of music production.

1. INTRODUCTION

Auditory masking is a well-known psychoacoustic phenomenon occurring when the presence of a loud sound compromises the perception of a fainter, adjacent sound by masking its presence. If the sounds are adjacent in terms of frequency components, the phenomenon is also called spectral masking, while if the sounds are temporally adjacent, it is called temporal masking [1–5]. In both cases, the amount of masking is a function of the distance between the two stimuli. By defining the distance measure and the masking function, different psychoacoustic models can be implemented, varying the degree of approximation according to the desired goals.

An example of a common psychoacoustic model is that used in the MPEG-2 audio compression [6]; a lossy compression technique which exploits a non-uniform quantization of the frequency content of a signal such that the quantisation noise remains under a certain Masking Threshold (MT), thus reducing the amount the information without affecting the perceived quality. The MP3 psychoacoustic model is devoted to the computation of the MT and can operate at different degrees of approximation, allowing a trade-off between encoding time and quality (not to be confused with bitrate).

In particular, the frequency masking contribution of a psychoacoustic model can account for a lot of different factors (loudness, frequency register, spectral complexity, and so on), but in its most simple definition, it says that the loudest frequency inside a critical band [7, 8] masks the fainter ones inside the same band. This phenomenon can be used to approximate an MT by convolution [9] or matrix multiplication [10] of the spectrum in the Bark scale [11] with a set of Spreading Functions (SF) – which is an asymmetrical triangular-like function the size of a critical band.

The goal of this work is to exploit such a mechanism to emphasize the masked frequencies of a signal in a real-time context, where the masker and the masked sounds are available as separate streams, such as tracks inside a Digital Audio Workstation (DAW). In the context of music production, this will allow the sound engineer to resolve the situation where an accompanying instrument masks a lead instrument [12], *e.g.* a guitar masking a vocal track.

To achieve this goal, a VST3 plugin called TheMasker has been implemented, taking as input the signal that will be processed and as a side-chain the signal that is potentially masking the input. An MT is computed from the side-chain, and then a dynamic equalizer boosts the input components lower than the MT. The user is provided with enough controls to fine-tune the process, but to avoid information overload, many algorithm parameters are not exposed, instead, they have been tuned empirically to adapt to most situations.

The novelty of the proposed approach lies in the use of a psychoacoustic model together with the possibility of managing the masking signal as a simple side chain input, which is a combination not sufficiently explored in the literature, as discussed in Section 2.

The context of music production has been chosen as a reference in order to ease testing (*i.e.* giving the plugin to professionals and collecting their feedback), but also other contexts may benefit from such processing, examples will be discussed in the conclusions.

The remainder of the paper is organized as follows: first, existing solutions will be examined in Section 2, then the implementation of TheMasker is described in Section 3, an evaluation campaign is reported and discussed in Section 4, and finally Section 5 concludes the paper with additional remarks and future works.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. EXISTING SOLUTIONS

The problem tackled in this context has already been discussed in the scientific literature, and also commercial products are already available on the market.

In general, a naive approach to the problem would be to use dynamic equalisers or multiband dynamics processors with the masker signal in the side chain to expand the portion of the masked signal whenever the masker is above a certain threshold. This threshold can be set manually (a general-purpose tool can be used in this sense) or automatically, such as in the work of Sack *et al.* [13], or in *MAuto-DynamicEq* [14]. Nevertheless, none of these approaches is based on a proper psychoacoustic model.

Finer processing may be achieved by working with a non-parametric representation of the target equalisation curve, such as in *TrackSpacer* [15], that uses a 32-band, fixed-frequency dynamic equaliser to apply an equalisation which corresponds to the flipped spectrum of the side chain input, but again, no explicit psychoacoustic model is mentioned.

Also Ahmetovic *et al.* [16] uses a non-parametric spectral representation, with a perceptually-motivated band choice in order to apply a gain to each band such that a target signal-to-noise ratio is reached.

A famous hardware processor that indeed accounts for psychoacoustic effects is the Vitalizer [17] (also available as VST plugin), which uses a patented model to achieve (among other things) a de-masking of those frequencies masked by other components of the same signal. Unfortunately, no side-chain is contemplated in this tool.

Gonzalez and Reiss [18] and Hafezi and Reiss [19] have proposed processors that use an elegant psychoacoustic model of interacting tracks; the first operating a broadband gain change of a target track, while the second implementing a sophisticated modulation of the parameters of a set of 5-band parametric equalisers on each examined track, minimizing their masking interactions. Note that this "multitrack" approach is more relevant in the context of automatic mixing, while constraining the intervention on a single track may be more suited for the realisation of a traditional mixing workflow tool.

The proposed solution is based on a simpler psychoacoustic model to that proposed by Gonzalez, Hafezi, and Reiss. Still, it uses a correction curve described by a 32-band filterbank, allowing finer control of the frequency content. Moreover, a two-input single-output routing scheme allows the integration in all those DAWs that do not support the multiple-input multiple-output routing required by Hafezi and Reiss.

3. IMPLEMENTATION

TheMasker has been first prototyped in Matlab and then implemented in C++ using the JUCE framework. In Figure 1 a block diagram of the signal processing strategy is visible: the side-chain signal is used to compute a frequency domain MT, and then the input spectrum is compared against the MT in a per-band logic. The difference in decibels of each band is used to control the gains of a Linkwitz-Riley band-pass filterbank according to the settings decided by the user.

Ideally, both the side-chain send and the plugin position in the channel processing chain should be post-fader, so to consider the actual mixing level, thus obtaining a realistic effect, nevertheless, the user is left with the ability to tweak side-chain, input, and output levels, exposed as plugin parameters.

A Graphical User Interface (GUI), visible in Fig. 2, has also been implemented to ease the testing campaign of the realised tool.

3.1 Masking Threshold Estimator

The MT is computed by taking a Flat-Top-windowed 1024-point 50%-overlap Fast Fourier Transform (FFT), resulting in 512 magnitude values of the side-chain input. Then, the resulting signal frequencies are spaced in the Bark scale by matrix multiplication with a 32×512 filterbank matrix, and converted in the Decibel scale. A reduction to a 32 values array is finally realised by matrix multiplication with a 32×32 SF matrix, providing a 32-points approximation of the side-chain MT.

The used SFs are expressed in dB and are centred on 32 linearly-spaced Bark frequencies f_b, with a rising slope of $+27$dB/Bark and a falling slope of -12dB/Bark [10, 20] that approximate the following SF expression from Painter and Spanias [21, 22]:

$$\mathrm{SF}(f_b) = 15.81 + 7.5(f_b + 0.474) - 17.5\sqrt{1 + (f_b + 0.474)^2} \tag{1}$$

an example of Eq. 1 is visible in Figure 3.

3.2 Input Correction Estimator

Similarly to the first stages of side-chain processing, the main input signal is taken to the 32-points Bark/dB space, but without any SF applied to it. This input representation is compared to the MT by means of a simple subtraction: positive delta values indicate that that band is masked by the side-chain, while negative delta values mean the input is not masked.

At this point, positive delta values should be directly used as gain dB values to ensure the masked input bands reach the MT, nevertheless, many issues may arise by doing so, therefore a number of constraints and corrections are enforced.

Silent or very quiet parts of the input spectrum would be amplified by a disproportionate amount than expected if in coincidence with a non-quiet part of the side-chain spectrum. This issue has been avoided by multiplying the delta values with an Intervention Limiting Function (ILF), computed as a function of the input representation. In particular, the ILF is computed as:

$$\mathrm{ILF}(n) := 0.5\left(1 + \tanh\left(\frac{x(n) - \tau}{k}\right)\right) \tag{2}$$

where $x(n)$ is the n-th band of the input representation, τ is a threshold value and k is a knee parameter. This function scales the delta values to 0 when the input is lower than τ,

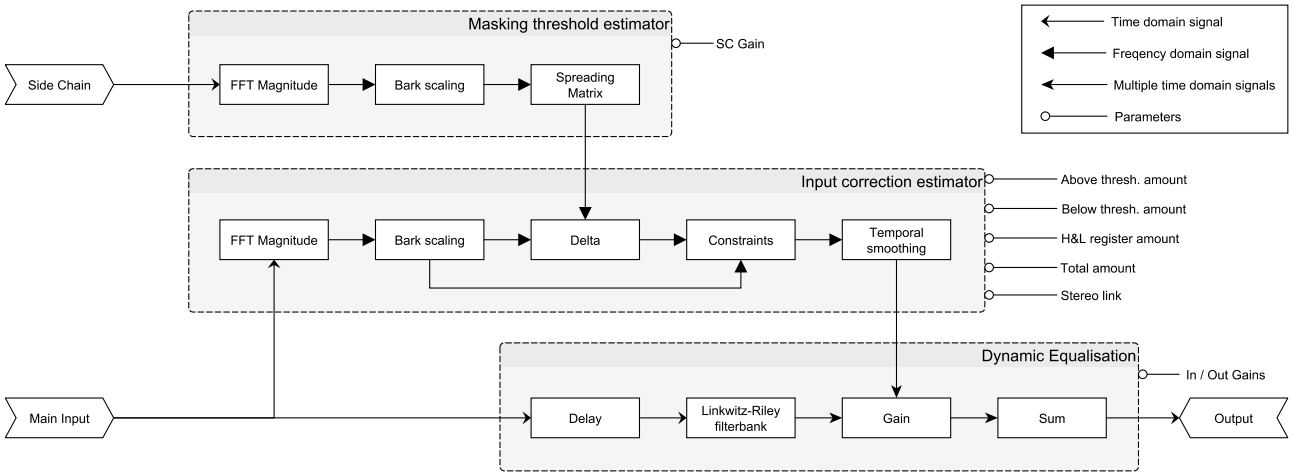


Figure 1. TheMasker block diagram.

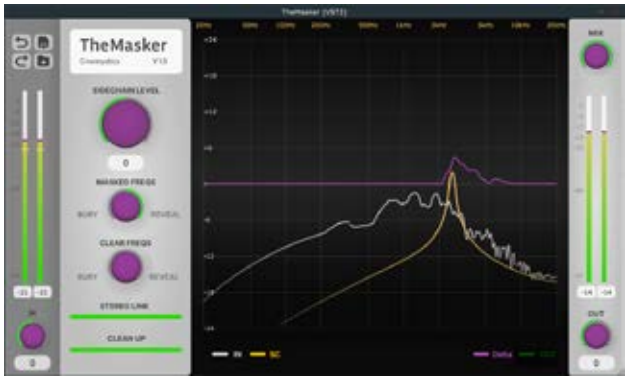


Figure 2. TheMasker Graphical User Interface. In this example, a sinewave masking some noise is visible, with the correction curve boosting all noise components below a spreading window.

and scales the delta values to their original value when the input is above τ. This behaviour is smoothed according to k, which ensures a gradual transition between the gated and the non-gated part. A τ of -40db with a $k = 3.2$ has been chosen since it produces a smooth transition between -50dB and -30dB, which has been empirically found to be adequate for most situations.

The same issue is actually present also in the case of the side-chain being very quiet, thus a second ILF with the

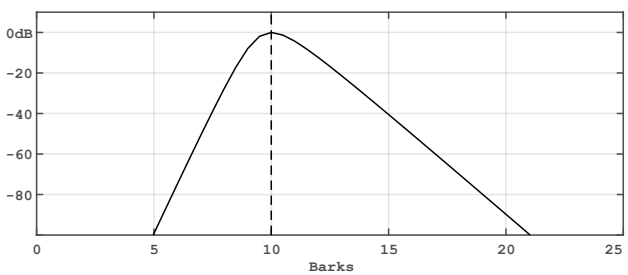


Figure 3. A spreading function centered on 10 Barks, as defined in Eq. 1

same τ and k parameters, but computed on the side-chain instead of the input signal, is applied to the delta values.

A second similar issue may arise even if the masked frequencies would genuinely benefit from a boost, but the amount of gain is so high that it may denature the sound in a perceivable way. For this reason, the delta values are soft-clipped between ± 12dB, again by means of an *tanh* function ($\delta(n)$ is the n-th band delta value):

$$\delta(n) = 12 \cdot \tanh\left(\frac{\delta(n)}{12}\right) \tag{3}$$

Finally, the delta values are converted to a correction equalisation curve by letting the user decide the scaling of the intervention in a number of ways:

- A scaling factor for the positive delta values, allowing the demasking of the input;
- A scaling factor for the negative delta values, allowing for the attenuation for the non-masked input components;
- A scaling factor for the overall delta values, to tune the overall intervention (exposed as an "amt" parameter);
- A frequency-dependent scaling factor allowing the user to process the whole spectrum or excluding the low and high register from being processed (exposed as a "Cleanup" parameter);

An additional "Stereo link" parameter allows the user to process the stereo channels independently or by averaging left and right delta values.

To avoid audible artefacts occurring in case of fast changes, the output values are smoothed in the time domain with different linear ramps for attack and release phases, set respectively to 80ms and 250ms. These values have been chosen empirically to maximise intervention transparency.

3.3 Dynamic Equalisation

First, to compensate for the latency introduced by the buffering needed to compute the FFT, a delay of 1024 samples is performed on the input signal. According to the VST3 standard, this delay is declared to the host application to let the DAW compensate for the introduced latency.

To achieve the dynamic equalisation the input signal is split into 32 bands via a Linkwitz-Riley crossover filterbank, with each band centred on the corresponding Bark scale representation frequency. Each band is then multiplied with the smoothed values provided by the Input Correction Estimator described in Section 3.2, compensating the cumulative effects of neighbour bands as described in [23], and then summed back to a single signal.

4. TESTING

4.1 Protocol

To test the effectiveness of TheMasker, a set of 6 stimuli has been administered to a pool of 13 subjects together with a questionnaire regarding intelligibility and perceived quality.

In particular, each stimulus is composed of three versions of two mixes. The two mixes are composed of the following signals:

- Masker: distorted guitar; masked: voice.
- Masker: long-term average speech spectrum (LTASS) noise [24]; masked: voice.

The first stimulus has been chosen as an ecological use case in the context of music production, while the latter is a synthetic example of speech enhancement in a noisy environment context.

The baseline version is an unprocessed mix of the tracks set to have the masked signal 3dB lower than the masker in terms of perceived loudness (the masker level is set to be -28 LUfs); in the second version the tracks are set to the same level of the baseline, but the masked signal is processed with TheMasker to let the masked frequencies reach the same level of the corresponding masker bands; finally, the last version is the same mix of the baseline but with a linear gain of $+3$dB on the masked signal.

The three versions of a single mix are presented together through closed headphones, with the possibility to switch between each version. For each mix, the subjects have to evaluate through a 5 points Likert scale the three versions according to the following questions:

1. How would you rate intelligibility?
2. Does the voice sound natural (1) or processed (5)?
3. How similar are the mixing levels if compared to this reference track?

For the final question, the user is given a new "reference" track which is a copy of the baseline. In this way, it is possible to check the reliability of a subject answers based on her ability to identify the baseline as identical to the reference.

The ideal outcome consists of TheMasker scores revealing an intelligibility improvement greater or equal to that of the simple volume increase correction, without too much loss in naturalness. Moreover, the version processed with TheMasker is expected to be more similar to the baseline in terms of mixing level similarity.

4.2 Results

First, it has been checked if some subjects failed to recognise the baseline and the reference as the same mix. It emerged that 3 subjects failed this task by giving a score ≤ 3 to the mix similarity of the baselines. These subjects were removed from the analysis, thus reducing the number of available subjects to 10 units.

Then a Kruskal-Wallis test was run for all 3 measures (intelligibility, naturalness and mix similarity) to see if answers relative to the 3 versions of the mixes (labelled Baseline, TheMasker, and Level in Fig. 4) are significantly different.

From the Kruskal-Wallis test, it emerged that indeed, intelligibility and mix similarity have been scored differently for the three versions ($p < 0.001$), while naturalness presents no statistically significant differences ($p = 0.087$).

Concerning intelligibility, Baseline is significantly less intelligible than the other two versions ($p < 0.005$) having a median score of 2, versus a median score of TheMasker and Level of 4 (TheMasker and Level have no significant differences, with $p = 0.768$)

The unexpected absence of significant differences in the perceived naturalness of sound (especially between Baseline and the other versions) can be interpreted as a kind of processing respectful of the original sound, which is understandable for the Level, but surprising for TheMasker.

Finally, the perception of mix similarity of TheMasker (median $= 3.5$) and Level (median $= 2$) resulted to be different ($p < 0.05$) in favour of the TheMasker, being evaluated more similar to the references than the Level version.

From these results, it can be deduced that TheMasker introduces intelligibility improvements comparable to those of a level increase, with the same impact on perceived timbre, but with less obvious changes in mix balance, thus demonstrating the usefulness of the proposed tool.

5. CONCLUSIONS

A prototype for the real-time compensation of psychoacoustic frequency masking has been implemented and tested in a real-world music-production scenario with promising results. The prototype is available for download at https://www.lim.di.unimi.it/demo_vst_eng.phpfor macOS, Windows, and Linux platforms.

Future works will focus on the improvement of the psychoacoustic model and the quality of the dynamic equalisation engine.

Also, other use cases of the proposed prototype will be

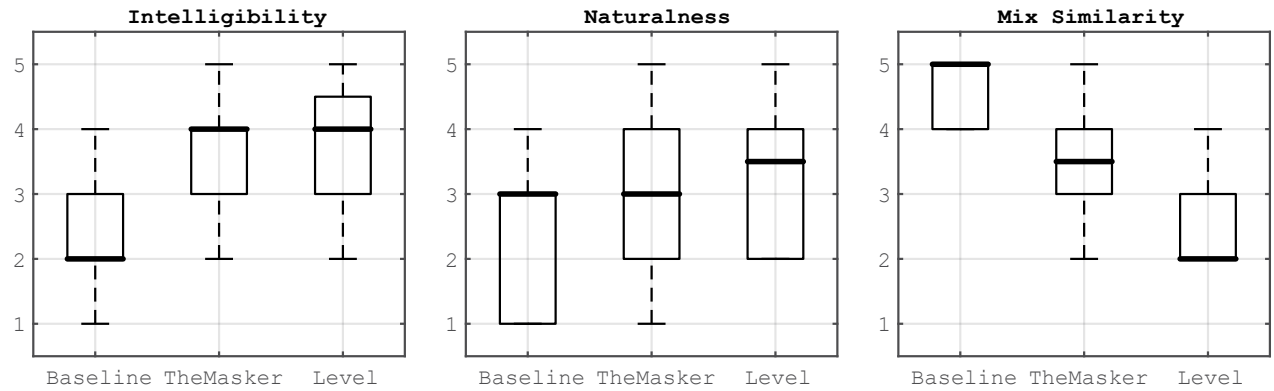


Figure 4. Results of the validation campaign: answers of 10 subjects when asked to compare a Baseline mix with two enhanced versions (one with a vocal track processed with TheMasker and one where the vocal track has only been amplified).

explored in the future, such as real-time intelligibility improvements of headphone output in noisy scenarios.

6. REFERENCES

[1] B. C. J. Moore, "Masking in the human auditory system," *Journal of the Audio Engineering Society*, May 1996.

[2] V. Best, "An introduction to psychological and physiological acoustics," *The Journal of the Acoustical Society of America*, vol. 150, no. 4, pp. A133–A134, 2021.

[3] E. Zwicker and H. Fastl, *Psychoacoustics: Facts and models*. Springer Science & Business Media, 2013, vol. 22.

[4] A. J. Oxenham, "Mechanisms and mechanics of auditory masking," *The Journal of physiology*, vol. 591, no. Pt 10, p. 2375, 2013.

[5] J. P. Egan and H. W. Hake, "On the masking pattern of a simple auditory stimulus," *The Journal of the Acoustical Society of America*, vol. 22, no. 5, pp. 622–630, 1950.

[6] I. J. S. 29, "Generic coding of moving pictures and associated audio information — part 3: Audio," Genève, Switzerland, Tech. Rep., 1998.

[7] H. Fletcher and W. A. Munson, "Loudness, its definition, measurement and calculation," *Bell System Technical Journal*, vol. 12, no. 4, pp. 377–430, 1933.

[8] H. Fletcher, "Auditory patterns," *Reviews of modern physics*, vol. 12, no. 1, p. 47, 1940.

[9] J. Herre and S. Dick, "Psychoacoustic models for perceptual audio coding—a tutorial review," *Applied Sciences*, vol. 9, no. 14, p. 2854, 2019.

[10] G. Schuller, *Psycho-Acoustic Models*. Springer International Publishing, 2020. [Online]. Available: https://doi.org/10.1007/978-3-030-51249-1_4

[11] E. Zwicker, "Subdivision of the audible frequency range into critical bands (frequenzgruppen)," *The Journal of the Acoustical Society of America*, vol. 33, no. 2, pp. 248–248, 1961.

[12] R. Izhaki, *Mixing audio: concepts, practices and tools*. Taylor & Francis, 2013.

[13] M. C. Sack, S. Buchinger, W. Robitza, P. Hummelbrunner, M. Nezveda, and H. Hlavacs, "Loudness and auditory masking compensation for mobile tv," in *2010 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, 2010, pp. 1–6.

[14] MeldaProduction. (2020) Mautodynamiceq. [Online]. Available: https://www.meldaproduction.com/MAutoDynamicEq

[15] Wavesfactory. (2014) Trackspacer. [Online]. Available: https://www.wavesfactory.com/audio-plugins/trackspacer/

[16] D. Ahmetovic, G. Galimberti, F. Avanzini, C. Bernareggi, L. A. Ludovico, G. Presti, G. Vasco, and S. Mascetti, "Enhancing screen reader intelligibility in noisy environments," *IEEE Transactions on Human-Machine Systems*, 2023.

[17] S. electronics. (2000) Vitalizer. [Online]. Available: https://spl.audio/en/spl-produkt/stereo-vitalizer-mk2/

[18] E. P. Gonzalez, J. D. Reiss *et al.*, "Improved control for selective minimization of masking using interchannel dependancy effects," in *11th Int. Conference on Digital Audio Effects (DAFx)*, 2008, p. 12.

[19] S. Hafezi and J. D. Reiss, "Autonomous multitrack equalization based on masking reduction," *Journal of the Audio Engineering Society*, vol. 63, no. 5, pp. 312–323, May 2015.

[20] G. Schuller. (2016) Psychoacoustics models. [Online]. Available: https://colab.research.google.com/github/GuitarsAI/AudioCodingTutorials/blob/master/AC_05_psychoAcousticsModels.ipynb

[21] T. Painter and A. Spanias, “A review of algorithms for perceptual coding of digital audio signals,” in *Proceedings of 13th International Conference on Digital Signal Processing*, vol. 1, 1997, pp. 179–208 vol.1.

[22] M. R. Schroeder, B. S. Atal, and J. Hall, “Optimizing digital speech coders by exploiting masking properties of the human ear,” *The Journal of the Acoustical Society of America*, vol. 66, no. 6, pp. 1647–1652, 1979.

[23] V. Välimäki and J. D. Reiss, “All about audio equalization: Solutions and frontiers,” *Applied Sciences*, vol. 6, no. 5, p. 129, 2016.

[24] A. Löfqvist, “The long-time-average spectrum as a tool in voice research,” *Journal of phonetics*, vol. 14, no. 3-4, pp. 471–475, 1986.

MODELING THE PULTEC EQP-1A WITH WAVE DIGITAL FILTERS

Alberto BARRERA[1], **Xavier LIZARRAGA-SEIJAS**[1], and **Frederic FONT**[1]

[1]**Universitat Pompeu Fabra**, Barcelona, Spain

ABSTRACT

This paper details the development of a virtual analog model of the passive equalizer stage of the classic Pultec EQP-1A studio equalizer using Wave Digital Filters. The modeling process involves analyzing the unit's schematics, LTspice simulations, and implementing a Wave Digital Filter structure using the `pywdf` library in Python. The proposed structure initially used *R*-Type adaptors, that are removed in a second model, better optimized for real-time. The Python prototype is compared to LTspice simulations, showing that, at sufficiently high sampling rates, the discrepancies between the two are marginal. The Wave Digital Filter model was then ported to C++ using the `chowdsp_wdf` library. Both implementations of the model in C++ (the one that uses *R*-Type adaptors and the real-time optimized one) are tested. Our optimized implementation proved to be much faster than the other open-source, Faust-based emulation of the EQP1-A that we evaluated, and a more accurate emulation of the original unit's equalizer stage.

1. INTRODUCTION

1.1 Motivation

Analog audio gear has been used in music production for almost a century and its distinct character continues to be sought after by many producers and sound engineers. Analog devices, including compressors, equalizers, and amplifiers, are highly regarded for their ability to introduce rich harmonic content, saturation and nonlinear characteristics into the processed signal.

Analog gear, however, can be fragile, needs maintenance, takes up physical space in the studio and is more expensive to acquire than its digital counterparts. It is also less convenient to use since it does not allow to recall configurations or use presets (like software does). During the last three decades, the processes involved in music recording, editing and post-processing have shifted to the digital domain and an important part of the audio equipment industry has disappeared, making these devices more and more scarce. Addressing these problems, virtual analog modeling techniques have long been of great interest to replicate the sound characteristics of analog gear in the digital environment [1–3].

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Virtual analog modeling methods can be categorized as black-box or white-box (or a mix of both, often referred to as grey-box). Black-box techniques consider the system as a whole while white-box approaches aim to model the system by focusing on the characteristics of its individual components and their interactions.

Black-box methods are usually better for capturing "unpredicted" behavior that is not reflected in the circuit schematic or where there is not a complete understanding of the mechanisms involved in the processing. As a disadvantage, control parameters do not always map to the model's parameters in obvious ways and a complex mapping is often needed. Some of the most common black-box techniques include swept-sine methods [4–6], Volterra Series [7, 8] or in the last few years, the use of Machine Learning and, specially, Deep Neural Networks [9–11].

White-box modeling, on the other hand, allow to accurately represent the system's behaviour from schematics even without having access to a real unit and allow for direct mapping of control parameters to specific components or variables of the model [12]. The development of these models require deep domain knowledge of physics, circuit theory and digital signal processing. The most common white-box modeling methods are Modified Nodal Analysis [13, 14] and Wave Digital Filters (WDF) [15]. While Modified Nodal Analysis remains extensively used for virtual analog modeling, the WDF approach has gained considerable recognition due to its simplicity, modularity and ease to adapt to real-time applications. These advantages, combined with the availability of software libraries for development and rapid prototyping of WDFs, makes it a very attractive method for modeling analog circuits [16, 17].

1.2 Wave Digital Filters

WDFs are a kind of digital filter based on physical modeling principles [15, 18]. They can be used to represent the physical state of an element, for example, the current physical state of a capacitor or an inductor. WDFs are also a kind of finite difference scheme with unusually good numerical properties.

In the WDF domain, each circuit element can be defined by a port with an incident and reflected wave and a free variable, the port resistance [15, 19]. The incident and reflected waves, a and b respectively, at a certain port are defined by equations (1) and (2). The port resistance, R_p can be set to any real value.

$$a = v(t) + R_p \cdot i(t) \tag{1}$$

$$b = v(t) - R_p \cdot i(t) \tag{2}$$

Where $v(t)$ and $i(t)$ are Kirchhoff's domain variables, the instantaneous voltage and current across and through the terminals of an element. Equations (3) and (4) are used to get $v(t)$ and $i(t)$ back from the WDF domain variables.

$$v(t) = \frac{a+b}{2} \tag{3}$$

$$i(t) = \frac{a-b}{2R_p} \tag{4}$$

In order to make the WDF structure, first it is necessary to discretize each of the components of the circuit. The usual method for this digitization is the bi-linear transform but other methods can be used. Table 1 details the resulting expressions for both the port impedance and reflected wave of common circuit elements

Table 1: Expression for the WDF's domain variables of the different circuit elements.

	R_p	b
Resistor	R	0
Capacitor	$1/(2f_sC)$	$z^{-1}a$
Inductor	$2f_sL$	$-z^{-1}a$
Voltage Source	Matches the rest of the circuit	$2v_s - a$

Where z^{-1} is a one sample delay and f_s is the sampling frequency.

To prevent instability due to delay-free loops when deriving the WDF elements, the reflected wave, $b[n]$, cannot be directly proportional to the instantaneous value of the incident wave, $a[n]$. However, it can be proportional to a delayed version of the incident wave, $a[n-\tau]$. Usually, choosing an appropriate value for R_p eliminates the possibility of such loops, as shown in [15].

Once all the components of the circuit have been individually discretized they can be connected using adaptors. Adaptors are two-port elements characterized by a reflection coefficient that determines what amount of each of the incident waves gets reflected into the connected components.

In traditional WDF theory it is only possible to connect elements using series or parallel adaptors; this limits the topologies that can be modeled to circuits in which all of the components are connected either in series or in parallel. The reflected waves to each of the elements connected to a WDF adaptor, b_1 and b_2, follow the expressions of table 2.

Table 2: Definition of the reflected waves for the WDF adaptors.

	b_1	b_2
Series	$-a_2 + \gamma \cdot (a_1 + a_2)$	$-a_1 + \gamma \cdot (a_1 + a_2)$
Parallel	$a_2 + \gamma \cdot (a_1 - a_2)$	$a_1 + \gamma \cdot (a_1 - a_2)$

Where γ is the reflection coefficient and a_1 and a_2 are the incident waves arriving from each element connected to the adaptor.

With all the circuit elements and adaptors defined, they are connected together in a SPQR tree structure [20], where all of the *Series*, *Parallel*, *Q* and *R* nodes are defined. At each computational cycle of a WDF algorithm it is necessary to first, recalculate and propagate the reflected waves towards the root of the tree. Then, the resulting incident wave is used to calculate the reflected wave at the root of the tree, which is then propagated downwards. Finally, the voltage values are calculated at the output of the circuit and values in reactive components are stored to be used in the next computational cycle.

Apart from the series and parallel topology limitation mentioned before, only one nonlinear element can be introduced into the circuit. To do so, the nonlinear element is placed at the root of the SPQR tree and all the branches are connected to it. To model circuits with complex topologies or circuits containing multiple nonlinear elements it is necessary to introduce special adaptors such as *N*-port or *R*-type adaptors [21], characterized by a scattering matrix that acts as a multidimensional reflection coefficient between all of the elements connected to the adaptor.

1.3 The Pultec EQP-1A Program Equalizer

The Pultec EQP-1A is a classic analog equalizer originally released in 1951 (as the EQP-1), becoming the first program equalizer to be introduced to the market. It was mainly used in radio stations to equalize the broadcast program. The unit has ever since become a favorite among audio professionals and has been used on innumerable recordings for mixing and mastering. It has also been cloned by different manufacturers and emulated as a digital processor by multiple audio plug-in companies.

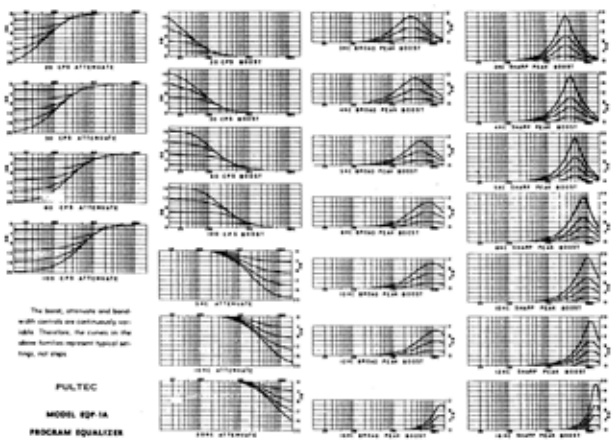


Figure 1: Frequency response curves from the EQP-1A user manual.

The EQP-1A consists of a passive equalizer stage and a vacuum tube gain stage to restore the insertion loss caused by the equalizer stage. The equalizer stage is composed by four different filters:

- Low frequency boost: shelving filter with selectable cutoff frequency and up to 16 dB of gain.
- Low frequency cut: shelving filter with up to 20 dB of attenuation.

- High frequency boost: resonant RLC filter with configurable center frequency and bandwith.
- High frequency cut: shelving filter with configurable cutoff frequency and up to 20 dB of attenuation.

The frequency response of each of the filters in different gain and frequency configurations from the original user manual are included in figure 1.

Although the EQP-1A's manual advises against boosting and attenuating simultaneously on the low frequencies, when both are applied at the same time, the boost and cut curves affect slightly different frequency bands. This allows to boost the low end, adding body to the sound while attenuating the low-mid frequency range to avoid "muddiness".

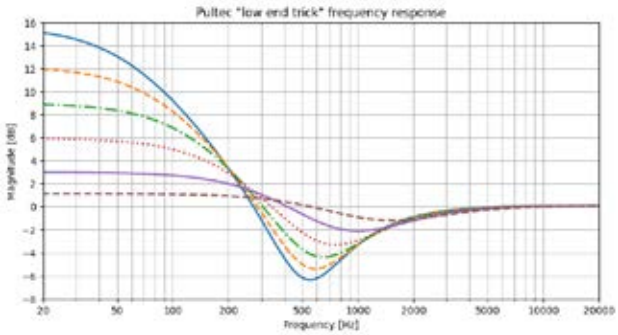


Figure 2: Frequency response of the EQP-1A when boosting and cutting the low frequency band.

Figure 2 illustrates the magnitude response of the equalizer when both boosting and attenuation are simultaneously applied at 60 Hz, with the boost and cut potentiometers set to the same position. This effect is commonly known as the "Pultec low end trick" and it is used extensively to enhance the sound of vocals, drums, guitars, or bass.

Because the equalizer stage is passive, it introduces signal loss effect. To make up for the 16 dB insertion loss, the signal level is restored using an output gain stage based on a 12AX7 and a 12AU7 vacuum tubes. Additionally, the original circuit incorporates input and output transformers. Nonetheless, this paper is focused on simulating the passive equalizer stage, with the acknowledgment that modeling the gain stage and input and output transformers of the unit is beyond its scope.

2. METHODOLOGY

2.1 References for the circuit

Since the original schematics for the circuit of the EQP-1A are not publicly available and we did not have access to the hardware unit, we took as reference a few documents that can be found online:

- The circuit diagram for the EQP-1R [22], another variant of the EQP-1A also built by Pulse Techniques.
- The open-source circuit diagram for the EQP-1A clone by Gyraf Audio [23].
- The frequency response curves from the EQP-1A user manual [24], already shown in figure 1.
- The frequency response curves from the EQP-1A 2019 reissue made by the original manufacturer taken from an article published in the Sound on Sound magazine [25] and measured using laboratory equipment.

Considering all of them we can gain a good understanding about the topology and components of the original device as well as its frequency response.

2.2 Circuit Simulations

Based on the references for the circuit, a first version of the circuit was simulated in LTspice. This first simulation showed some discrepancies with the measurements taken from the 2019 reissue of the unit so the value of some components was adjusted to closely match them.

Since the model does not need to adapt the input and output impedance of the circuit, the input and output resistors are removed, adjusting the necessary components to maintain the circuit's behavior. This would allow us to model the circuit in the WDF domain without the need of *R*-Type adaptors, making the processing faster for the same circuit's response. The version of the circuit used for the final model is included in figure 3.

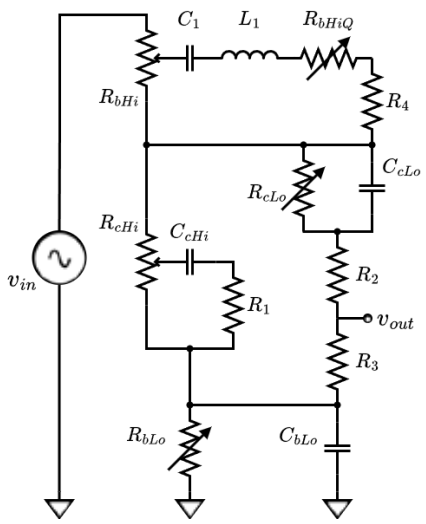
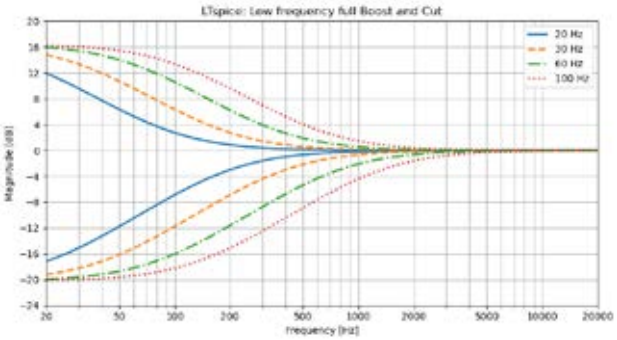


Figure 3: Modified version of the circuit used for the final model.

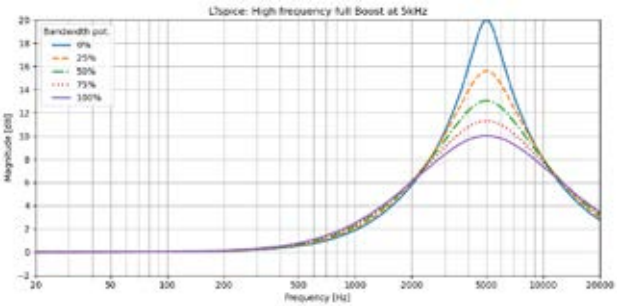
The results of the simulations carried out on this modified version of the circuit are included for: the maximum low frequency boost and cut configurations for all of the selectable frequencies (figure 4a), different bandwidth values for the resonant high frequency filter at maximum boost at 5 kHz (figure 4b) and maximum cut for all of the selectable frequencies of the high shelving filter (figure 4c).

All of the simulations were carried out with the same configuration as the measurements taken from the hardware

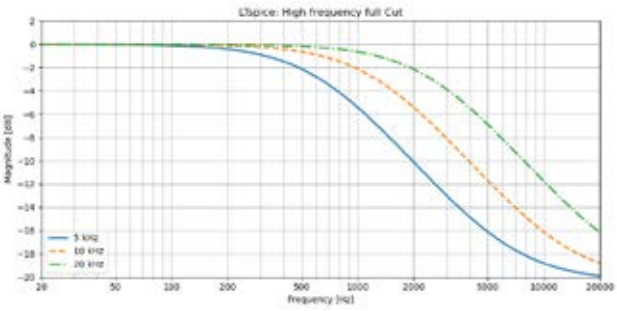
unit in the Sound and Sound article [25] so they could be directly compared against them. Although they could not be compared numerically in a meaningful way, the achieved response curves are identical.



(a) Low frequency boost and cut



(b) High frequency boost at 5kHz



(c) High frequency cut

Figure 4: LTspice simulations frequency responses

2.3 Wave Digital Filter Implementation

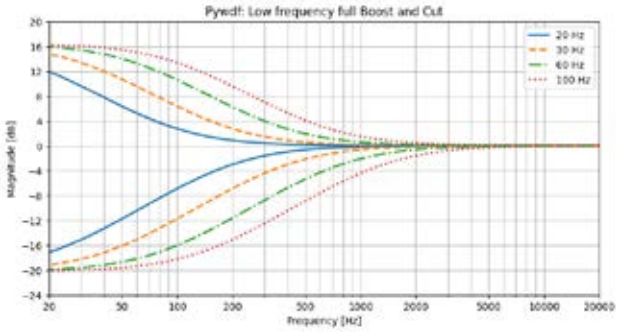
Once the frequency response of the simulations closely matches the references, the circuit is constructed in the WDF domain using Python and the `pywdf` library [1] [16]. `pywdf` includes WDF models for resistors, capacitors and inductors, as well as series, parallel and *R*-Type adaptors.

A first implementation of the circuit, that used *R*-Type adaptors, was developed following the circuit of the references. To derive the *R*-Type adaptor, the circuit's nodes and ports are labeled to create a net-list that describes the circuit. With this net-list the scattering matrix can be computed using *R*-Solver and SageMath. The process of obtaining the scattering matrix from the circuit is explained in an article published online by Jatin Chowdhury [26].

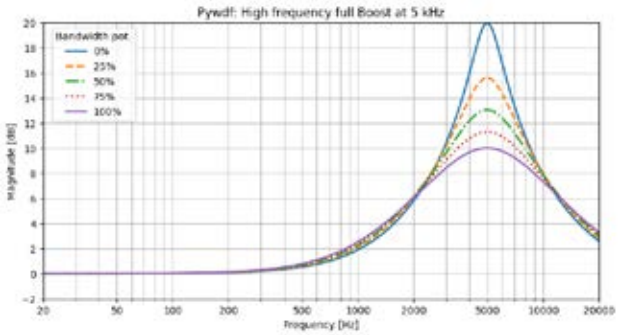
This first model, that uses *R*-Type adaptors, had the same response as the final version but worse performance. The performance of both models is compared in section 3.2.

The final WDF implementation of the circuit followed the circuit on figure 3, which resulted in the WDF structure represented in figure 6. The voltage source is connected at the root of the circuit and all of the components are connected through series and parallel adaptors to it.

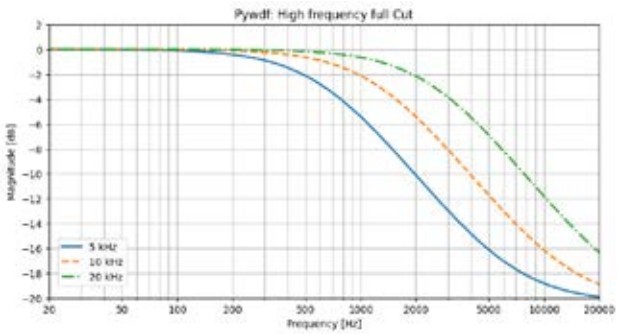
Using the `pywdf` model and the functions included in the library we can easily obtain the frequency response of the circuit. The frequency responses of the model for the same configurations as in the simulations carried out in LTspice and the ones measured from the hardware unit were computed at a sampling rate of 192 kHz and are included in figures 5a, 5b and 5c.



(a) Low frequency boost and cut



(b) High frequency boost at 5kHz



(c) High frequency cut

Figure 5: WDF model frequency responses

The frequency responses obtained are very accurate at

[1] `https://github.com/gusanthon/pywdf`

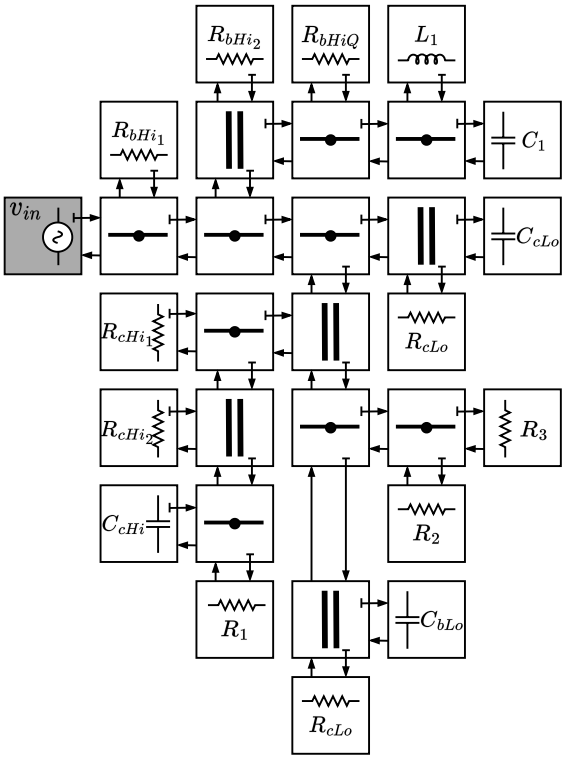


Figure 6: WDF diagram of the passive equalizer stage.

high sampling rates, but the Python models are slow and only able to process audio offline. While Python and `pywdf` provides a valuable prototyping environment, its limitations become apparent in real-time applications where faster programming languages, like C++, are needed.

2.4 Real-time C++ Model

After achieving accurate results from the WDF model in Python, it is necessary to translate the WDF model to C++ to enable real-time operation. To achieve this we used the `chowdsp_wdf` library [2] [17] for the WDF modeling and the JUCE framework [3] to compile the model into an audio plug-in.

One big advantage of using `pywdf` library for prototyping in Python alongside the `chowdsp_wdf` library for the real-time implementation is that they share the same foundation: all the processes are equivalent and numerically they shed the same results. This ensures that a model prototyped in Python can be seamlessly ported to C++ without any alterations, preserving identically the model's topology and behavior.

At lower sampling rates, and due to the use of the bilinear transform during the discretization of the circuit elements, considerable cramping occurs on the higher end of the spectrum. For this reason, getting good accuracy out of the WDF model (as quantified in section 3.1) requires the use of high sampling rates above the conventional 48 kHz used in audio production. However, running an entire project at high sampling rates is not always efficient.

[2] `https://github.com/Chowdhury-DSP/chowdsp_wdf`
[3] `https://juce.com`

To address this, our plug-in performs a series of operations: it internally up-samples the input signal, processes the over-sampled signal using the WDF model, and then down-samples the processed signal back to the original host's sampling frequency. The plug-in provides options for x2, x4, or no oversampling.

The result of this process is a fully operational audio plug-in that effectively incorporates the WDF model within a real-time environment. The plug-in can be used in any Digital Audio Workstation and it is available to download on GitHub [4]. A snapshot of its GUI is included on figure 7.

Figure 7: Graphical user interface of the plugin.

3. RESULTS

3.1 Frequency Response

To asses the accuracy of our WDF implementation, the frequency response curves computed from the Python model are compared against the ones obtained through LTspice simulations and the error between both is calculated.

For each graph, both the frequency response obtained from LTspice and the one computed from `pywdf` are displayed. The background color indicates the squared error between the two, $(|H_{\texttt{LTspice}}(f)| - |H_{\texttt{pywdf}}(f)|)^2$, and the $RMSE$, that follows the definition of equation (5), is included at the bottom left corner.

$$RMSE = \sqrt{\frac{1}{N}\sum_k (|H_{\texttt{spice}}(k\Delta f)| - |H_{\texttt{wdf}}(k\Delta f)|)^2} \quad (5)$$

The error graphs for an example EQP-1A configuration are included in figures 8, 9 and 10 for sampling frequencies of 48 kHz, 96 kHz and 192 kHz respectively. The numerical values for the error of the individual filters at the the most adverse configurations are gathered in table 3.

Table 3: Magnitude response error on each filter at the most adverse configuration.

	LF Boost	**LF Cut**	**HF Boost**	**HF Cut**
fs (kHz)	**RMSE**	**RMSE**	**RMSE**	**RMSE**
48	3.65e-2	2.07e-3	1.40	3.60e-2
96	1.86e-2	5.43e-4	6.44e-1	1.28e-2
192	9.37e-3	1.39e-4	2.20e-1	3.96e-3

[4] `https://github.com/ABSounds/EQP-WDF-1A`

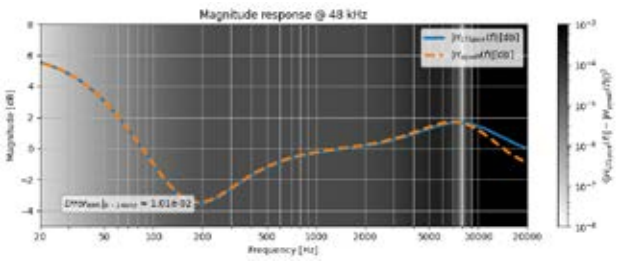


Figure 8: Example configuration frequency response error at 48 kHz.

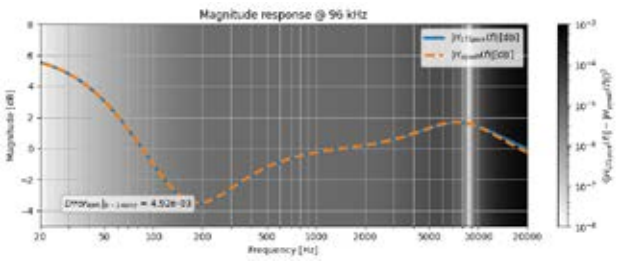


Figure 9: Example configuration frequency response error at 96 kHz.

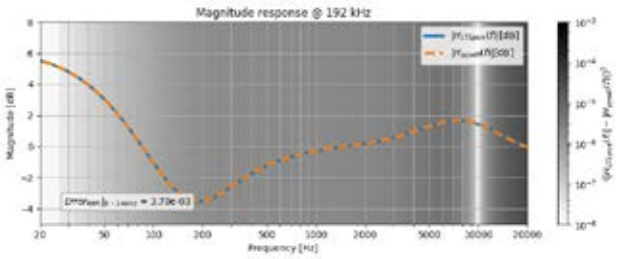


Figure 10: Example configuration frequency response error at 192 kHz.

The comparisons show that at higher sampling rates the model is very accurate and the error between the simulations and the WDF model are minimal.

Although we focused on the model's magnitude response in this work, we also examined its phase response and found no significant discrepancies from the expected behavior of the unit. The phase response does exhibit distortion when approaching the Nyquist frequency; however, as with the magnitude response error, this distortion can be reduced in the audio range by processing at higher sample rates.

3.2 Performance

To assess the performance of the different models, a random audio spanning 20 seconds is generated and processed using each model. The time taken to process all samples is measured using Python's `time.monotonic()` function from the standard library `time`.

The Python implementations are executed directly on a Jupyter Notebook, while the VST3 plug-ins are loaded into Python using Spotify's Pedalboard library [5] . All the processing is executed locally on a computer with an i7-8550U processor running at 1.8 GHz and 16 GB of RAM operating on Windows 11.

The evaluation includes the processing time for the 20 seconds of audio, as well as a real-time ratio. This ratio indicates how many times faster than real-time the model can operate. For instance, a ratio of 2 implies that the model is able to process 2 seconds of audio in 1 second. Any ratio below 1 means that the model is not suitable for real-time use.

Table 4 includes the comparison between the performance of our two Python models developed using the `pywdf` library. The first one uses traditional WDF with the modified version of the circuit, while the other one uses *R*-Type adaptors.

Table 4: Time taken to process 20 s of audio with each of the Python models.

	pywdf		**pywdf w/ R-Type**	
fs (kHz)	**Time (s)**	**Ratio**	**Time (s)**	**Ratio**
48	**26.969**	**0.74**	120.775	0.17
96	**53.931**	**0.37**	229.953	0.09
192	**108.984**	**0.18**	451.212	0.04

The results show that the implementation that used *R*-Type adaptors was 75% slower than the final one. Working at typical audio rates (48, 96 and 192 kHz), none of them was able to run in real-time. This was expected and caused by the nature of the Python programming language.

Table 5 includes the performance assessment of our C++ model compiled into a VST3 plug-in (EQP-WDF-1A) and compares it to another implementation of the EQP-1A described on this paper [27] developed in the Faust programming language that uses *R*-Type adaptors and was also compiled to VST3.

Table 5: Time taken to process 20 s of audio with each of the VST3 plug-ins.

	EQP-WDF-1A		**Faust EQP-1A (R-Type)**	
fs (kHz)	**Time (s)**	**Ratio**	**Time (s)**	**Ratio**
48	**0.092**	**217**	0.175	114
96	**0.182**	**110**	0.326	61
192	**0.360**	**56**	0.638	31

Overall, our plug-in performed between 40% to 47% faster than the Faust implementation on every configuration.

[5] `https://github.com/spotify/pedalboard`

4. CONCLUSIONS

Based on our measurements we can conclude that our WDF model accurately replicates the frequency response of the Pultec EQP-1A, specially at higher sampling rates, as confirmed by the comparisons between the LTspice simulations, the Python WDF model frequency response and measurements taken from the 2019 reissue of the unit.

The C++ implementation of the WDF model is also efficient and capable of real-time operation. It performs faster than existing implementations of the circuit and yields better results. Thanks to the use of the `pywdf` and `chowdsp_wdf` libraries, transitioning from the Python prototype to a real-time C++ implementation is a smooth process that preserves the model behaviour intact with some minor adaptations to the Python code.

This development was a first step in the direction to a complete open-source model of the Pultec EQP-1A. Although the frequency response achieved is faithful to the original unit, other sections of the circuit (specially the gain stage and input/output transformers) would need to be modeled to recreate its harmonic properties.

4.1 Future work

Various methods can be used to model the distortion produced by the 12AX7 and 12AU7 vacuum tubes in the EQP-1A's gain stage.

One option would be to use a filtering block and a static waveshaper as proposed in [28, 29]. This is a simple to implement but won't fully model the non-symmetrical distortion present in circuits with pairs of triodes like this one.

A more accurate option would involve integrating vacuum tube models into a WDF implementation, as discussed in [30, 31]. This would offer a high accuracy but might prove challenging due to the complex interaction between the gain and equalizer stages.

Instead of the WDF models of the vacuum tubes, neural networks can be introduced into the WDF domain model as proposed in [32]. This has shown great results in modeling guitar amplifiers and would likely work as well for the gain stage of the EQP-1A but it would not reduce the complexity of the WDF model.

Acknowledgments

This research was carried out under the project "IA y Música: Cátedra en Inteligencia Artificial y Música" (TSI-100929-2023-1), financed by Secretaría de Estado de Digitalización e Inteligencia Artificial, and Unión Europea-Next Generation EU, under the program Cátedras ENIA 2022. This paper is also based on the research conducted during a Master's Thesis on Sound and Music Computing at the Universitat Pompeu Fabra in Barcelona. Our gratitude goes to Audio Merge's [6] team for leading this proposal and to Xavier Lizarraga and Frederic Font, members of the Music Technology Group at UPF, for guiding. Their support and expertise were key to the success of this research.

[6] `https://audio-merge.com`

5. REFERENCES

[1] S. D'Angelo, "Lightweight virtual analog modeling," in *Proceedings of the 22nd Colloquium on Music Informatics, Udine, Italy*. Independent Researcher, 2018, pp. 20–23.

[2] A. Bernardini, A. Sarti *et al.*, "Towards inverse virtual analog modeling," in *Proc. 22nd International Conference on Digital Audio Effects (DAFx 2019)*, 2019, pp. 1–8.

[3] V. Valimaki, J. D. Parker, L. Savioja, J. O. Smith, and J. S. Abel, "Fifty years of artificial reverberation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 5, pp. 1421–1448, 2012.

[4] A. J. Berkhout, D. de Vries, and M. M. Boone, "A new method to acquire impulse responses in concert halls," *The Journal of the Acoustical Society of America*, vol. 68, no. 1, pp. 179–183, 1980.

[5] A. Farina, A. Bellini, E. Armelloni *et al.*, "Non-linear convolution: A new approach for the auralization of distorting systems," *Preprints-Audio Engineering Society*, 2001.

[6] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Audio engineering society convention 108*. Audio Engineering Society, 2000.

[7] T. Hélie, "Volterra series and state transformation for real-time simulations of audio circuits including saturations: Application to the moog ladder filter," *IEEE Transactions on Audio, Speech Language Processing*, vol. 18, pp. 747–759, 2010.

[8] A. Farina, E. Armelloni *et al.*, "Emulation of not-linear, time-variant devices by the convolution technique," in *Congresso AES Italia, Como*, 2005.

[9] J. Chowdhury, "A Comparison of Virtual Analog Modelling Techniques for Desktop and Embedded Implementations," Sep. 2020, arXiv:2009.02833 [cs, eess]. [Online]. Available: http://arxiv.org/abs/2009.02833

[10] A. Wright, E.-P. Damskägg, and V. Välimäki, "Real-Time Black-Box Modelling with Recurrent Neural Networks," in *Proceedings of the International Conference on Digital Audio Effects*. United Kingdom: University of Birmingham, Sep. 2019.

[11] C. J. Steinmetz, J. Pons, S. Pascual, and J. Serrà, "Automatic multitrack mixing with a differentiable mixing console of neural audio effects," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 71–75.

[12] M. Verasani, A. Bernardini, and A. Sarti, "Modeling sallen-key audio filters in the wave digital domain," in *2017 IEEE international conference on acoustics,*

speech and signal processing (ICASSP). IEEE, 2017, pp. 431–435.

[13] C.-W. Ho, A. Ruehli, and P. Brennan, "The modified nodal approach to network analysis," *IEEE Transactions on Circuits and Systems*, vol. 22, no. 6, pp. 504–509, Jun. 1975.

[14] K. Dempwolf, M. Holters, and U. Zölzer, "Discretization of parametric analog circuits for real-time simulations," in *Proceedings of the 13th international conference on digital audio effects (DAFx'10)*, 2010.

[15] A. Fettweis, "Wave digital filters: Theory and practice," *Proceedings of the IEEE*, vol. 74, no. 2, pp. 270–327, Feb. 1986.

[16] G. Anthon, X. Lizarraga-Seijas, and F. Font, "Pywdf: an open source library for prototyping and simulating wave filter circuits in python," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx-23)*, Denmark, 2023, pp. 335–341.

[17] J.Chowdhury, "chowdsp_wdf: An Advanced C++ Library for Wave Digital Circuit Modelling," Oct. 2022, arXiv:2210.12554 [eess]. [Online]. Available: http://arxiv.org/abs/2210.12554

[18] J. O. Smith, *Physical Audio Signal Processing*, ch. Wave Digital Filters, online book, 2010 edition. [Online]. Available: https://ccrma.stanford.edu/~jos/pasp/Wave_Digital_Filters_I.html

[19] multivac61, "multivac61/wave_digital_notebook," May 2023, original-date: 2016-10-23. [Online]. Available: https://github.com/multivac61/wave_digital_notebook

[20] K. J. Werner, "Virtual Analog Modeling of Audio Circuitry Using Wave Digital Filters," Ph.D. dissertation, Stanford University, Stanford, 2016, pages: 261 Volume: Ph.D. in Computer-Based Music Theory and Acoustics.

[21] K. J. Werner, V. Nangia, J. O. Smith III, and J. S. Abel, "Resolving wave digital filters with multiple/multiport nonlinearities," in *Proc. 18th Conf. Digital Audio Effects*, 2015, pp. 387–394.

[22] "Dave Library." [Online]. Available: http://www.davegroupjapan.com/ekoukoku9.html

[23] Gyraf Audio, "The G-Pultec." [Online]. Available: https://www.gyraf.dk/gy_pd/pultec/pultec.htm

[24] Pulse Techniques, "Pultec EQP-1A User Manual." [Online]. Available: https://www.thehistoryofrecording.com/Manuals/Pultec/Pultec_EQP-1A_Manual.pdf

[25] H. Robjohns, "Pulse Techniques EQP-1A," *Sound on Sound*, Feb. 2019. [Online]. Available: https://www.soundonsound.com/reviews/pulse-techniques-eqp-1a

[26] J. Chowdhury, "Wave Digital Circuit Models with R-Type Adaptors," Oct. 2022. [Online]. Available: https://jatinchowdhury18.medium.com/wave-digital-filter-circuit-models-with-r-type-adaptors-39ad0ad658ce

[27] D. Roosenburg, E. Stine, R. Michon, and J. Chowdhury, "A wave digital filter modeling library for the faust programming language," in *Proc. 18th Sound Music Comput. Conf., Virtual*, 2021, pp. 24–30.

[28] R. Kuehnel, "Digital Modeling of Guitar Amplifier Preamp Distortion." [Online]. Available: https://www.ampbooks.com/mobile/dsp/preamp/

[29] I. Cohen, "Fifty shades of distortion," in *ADC'17*, London, UK, Nov. 2017. [Online]. Available: https://www.youtube.com/watch?v=oIChUOV_0w4

[30] M. Karjalainen and J. Pakarinen, "Wave digital simulation of a vacuum-tube amplifier," in *2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings*, vol. 5. IEEE, 2006.

[31] J. Pakarinen and M. Karjalainen, "Enhanced wave digital triode model for real-time tube amplifier emulation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 4, pp. 738–746, 2009.

[32] C. C. Darabundit, D. Roosenburg, and J. O. Smith III, "Neural net tube models for wave digital filters," in *Proc. 25th Int. Conf. Digital Audio Effects (DAFx20in22)*. Vienna University of Music and Performing Arts, 2022, pp. 153–160.

REVITALIZING MULTIMEDIA ART: RESTORATION AND RE-ACTIVATION OF AN AUDIOVISUAL INSTALLATION IN VIRTUAL REALITY

Alessandro RUSSO (alessandro.russo@dei.unipd.it) (0000-0001-6691-759X) [1],
Andrea FRANCESCHINI (andrea.franceschini@dei.unipd.it) (0000-0001-8665-6147) [1],
Antonio RODÀ (roda@dei.unipd.it) (0000-0001-9921-0590) [1], and
Sergio CANAZZA (sergio.canazza@unipd.it) (0000-0001-7083-4615) [1]

[1] *Centro di Sonologia Computazionale, Department of Information Engineering*, **University of Padova**, Italy

ABSTRACT

In this paper we describe the re-activation of the multimedia artwork "La(M)pelle di Ahmad" (1979) by Italian artist Roberto Taroni, an installation that combines visual and sonic arts, using Super 8 films and magnetic audio tapes. After the digitization and the digital restoration of both film and audio components, we re-activated the artwork in a Virtual Reality (VR) environment. This technology provides a unique opportunity to recreate the immersive experience, and to preserve the original concept of the artwork, enabling the audience to experience it in a digital context. We recreated the physical features of the original installation in the virtual environment, as well as the interplay between the multiple audio and film sources, aiming for a faithful recreation of the original installation that also incorporates considerations from the artist. Maximizing access to the installation was one of the main objectives during the design of the digital experience, therefore we took care to balance the needs for faithful preservation and for usability on older, more widely available hardware. This project revitalized the historical artwork, underscoring VR's efficacy in preserving cultural heritage in resource-constrained settings by exploring the possibilities offered by virtual reality in preserving and re-activating multimedia art installations.

1. INTRODUCTION

In contemporary arts, the coalescence of diverse media, aided by advances in interactive technologies, has become a defining characteristic of modern artistic expression. Art forms can be characterized by the technologies employed (e.g., artificial intelligence, virtual reality), multimedia (e.g., film, video, sound), or the modes of engagement (e.g., installation, performance, interaction). In addition, the temporal and spatial context assumes a central importance in the creation of the artwork, alongside the dynamic nature it may assume due to internal or external influences (e.g., audience interaction). New multimedia artworks diverge significantly from traditional fixed forms such as paintings, sculptures, and architecture, because they lack permanence. Multimedia artworks, whether film, video, sound, or computer-based, need innovative preservation methodologies and new access-oriented approaches tailored to the unique characteristics of contemporary art.

Roberto Taroni emerges as a vibrant presence within the artistic scenario. Born in 1957, since the inception of his career, he has embarked on an ambitious and intricate endeavor, crafting a holistic artistic vision that traverses various disciplines, combining sound (especially electronic music) and visual arts. Taroni's artistic journey is distinctive in its exploration of diverse modes of expression, intricately weaving together a tapestry of influences and stylistic approaches. Over time, he has established himself as a fixture in the art world, evolving as a visual artist and a distinctive designer. His artistic pursuits encompass a broad spectrum of forms, including multimedia installations, films, performances, and *expanded cinema* [1], adding layers of complexity to his work. The exploration of expanded cinema holds particular significance within Taroni's body of work, as it not only pushes the boundaries of traditional filmmaking but also creates immersive and multisensory (visual and aural) experiences for audiences within his installations.

Taroni's artistic collection includes a vast array of cinematographic works and a rich archive of sound materials, often used by the artist in his installations and performances. The Centro di Sonologia Computazionale (CSC) [2, 3] of the University of Padova worked on the digitization and restoration of the whole corpus of sound materials, including 1/4" open-reel tapes (stereo, quadriphonic, and 8 channels), traditional audiocassettes, as well as the less common "Endless" type, used for creating small loops (usually between 3 and 5 minutes, and up to 12 minutes long), examples of which are shown in figure 1.

This digitization intervention followed a previous digitization of film materials by Taroni-Cividin, conducted by La Camera Ottica Film and Video Restoration Lab of the University of Udine in 2022-2023, when the first author of this paper was part of the preservation team. This standalone preservation project was carried out in relation to *Taroni-Cividin. Performance, Video, Expanded Cinema (1977-1984)* [4], supported by the Directorate-General for Contemporary Creativity - Italian Ministry of Culture under the Italian Council program (2021).

Starting from 2023, the research team of the CSC

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Figure 1. Original audio recordings on 1/4" open reel tapes and audiocassette used in the installations.

worked with the artist in exploring new possibilities for re-activating his artworks and digitally restoring the digitized materials, focusing on a particular case study: the 1979 installation *La(m)pelle di Ahmad*.

2. MOTIVATION AND RELATED WORKS

Traditional methodologies employed by archives and museums, while effective for many forms of art, often prove insufficient for preserving and re-activating multimedia art installations. In fact, multimedia art installations present unique challenges in terms of preservation and accessibility, often related to their complex nature (i), the short-life expectancy and obsolescence of digital and analog components (ii), and their time and process-based nature (iii). Nevertheless, the tendency of traditional paradigms is to preserve installations as fixed objects, rather than dynamic ones. Experimental preservation strategies have been applied to revive numerous multimedia pieces. Since the 2000s, several European initiatives, such as those undertaken by ZKM in Karlsruhe and the Archive of Digital Art (ADA), and projects such as the Horizon's *Dynamic Preservation of Interactive Art: The next frontier of Multimedia Cultural Heritage* and *New Approaches in the Conservation of Contemporary Art* (NACCA) have focused on preserving new multimedia art, but there is still the need of a shared preservation methodology.

Another open issue concerns not only the technical and material aspects of the artwork but also the *user experience* and the *interaction* within the surrounding space, since multimedia installations often engage viewers in an interactive experience with the environment.

To overcome these challenges, the CSC started to work on this topic in 2014 [5], and this effort led to the definition of the Multi-Dynamic Preservation Model (MDP) described in [6]. To refine and expand the MDP model, the CSC research team is currently exploring new re-activation strategies and considering new case studies. The advent of modern technologies, such as Virtual and Augmented Reality, offers a promising solution to this challenge. Virtual Reality enables the virtualization of real environments [7] and creates interesting possibilities for re-activating obsolete multimedia installations in an immersive and interactive manner, allowing users to fully experience the artwork. The recent progress in VR technology expanded its applicability across various domains. The emergence of relatively affordable hardware from corporations like HTC and Oculus has made VR more accessible, alongside software platforms often coming from the video game industry, which underscores the continued influence of the gaming sector on VR development [8].

In recent years, the possibilities offered by VR technologies in the field of preservation started to be explored by some international projects, for example *Obsolete Equipment*, a collaboration of the Netherlands Media Art Institute (NIMk) and PACKED, Platform for the Archiving and Preservation of Audiovisual Arts, together with MHKA Museum van Hedendaagse Kunst Antwerpen (Museum for Contemporary Art Antwerp), and S.M.A.K. Stedelijk Museum voor Actuele Kunsten Ghent (Municipal Museum of Contemporary Art in Ghent), SMA Stedelijk Museum Amsterdam, Museum Kröller-Müller en Instituut Collectie Nederland [9].

In this scenario, the possibilities offered by VR may also be explored in the field of digital restoration and its related documentation. The use of this technology may in fact allow the users to visualize the installation before and after the restoration intervention. This can be particularly interesting when dealing with multimedia installations that use analog *time-based media* [10] such as videos, open-reel magnetic tapes, films, and audiocassettes as fundamental components of the artwork. These materials face several problems related to obsolescence and physical/chemical degradation of the carriers, which significantly lower their life expectancy and the quality of content. Before restoration, VR enables the visualization of the artwork in its original state, allowing the users to experience its original preservation conditions. After restoration, VR allows the direct comparison of the previous state with the current one, highlighting the transformations and improvements introduced by the intervention.

3. CASE STUDY: LA(M)PELLE DI AHMAD

Among Taroni's works, the 1979 installation *La(m)pelle di Ahmad* stands out as a significant case study for its use of space, multi-projection, and audio/video looping as a means to explore new artistic possibilities. This installation is an emblematic example of multisensory and conceptual exploration that gives the users an immersive experience. Comprising three films projected onto as many walls, *La(m)pelle di Ahmad* immediately captures the viewer's attention. The first projection, taken from the film *Umwelt* (1978), strikes with its simplicity and evocative power: a light bulb suspended in darkness, the thin filament disappearing into the void, evokes a sense of mystery and contemplation. The second film, extracted from *La bocca di Ahmad* (1978) focuses on the mouth itself, a symbol of communication and introspection, opening and closing in a hypnotic rhythm, inviting the viewer to reflect on the meaning of language and human expression. Lastly, the third fragment unveils an additional layer of complexity, depicting the skin on the back of a knee where the

schematic for the installation is drawn. This metaphorical representation of the structure of the artwork questions the relationship between the artist, the artwork, and the users, opening spaces for an interpretation that extends far beyond the visible surface of the installation.

The film components are accompanied by three different audio loops that create a surreal dimension of interaction between sound and visual components. The loops were recorded on a tape used by the artist during the performances. In fact, in the whole of Taroni's artistic production, the sound element plays a very central role, almost as much as the visual content.

4. DIGITIZATION AND DIGITAL RESTORATION

4.1 Film materials

In 2022, the research and technical team of the Camera Ottica Lab, University of Udine, carried out a first digitization of the films belonging to the artist's collection. The materials comprised different films shot in 16 mm, 8 mm, and Super8 formats. The digitization ensured the creation of 2K digital preservation copies, as well as the inspection and reparation of the original materials. Materials filmed in 8 mm and Super8 were digitized using an MWA Choice film scanner, while the 16 mm films were digitized using the Blackmagic Design Cintel film scanner; the files were encoded with Apple ProRes 422.

Figure 2. Digitized Super8 films used in the installation before color correction intervention (a) and after (b).

The original Super8 films used for *La(m)pelle di Ahmad* were contained in special cassettes that allowed the artist to play their content in an endless loop. Therefore, before proceeding with the inspection and reparation of the material, it was necessary to extract the film from the cassette box. Some of the digitized films exhibited severe color fading, typically in magenta tones, which required correction. This was carried out by the research team of CSC using the software DaVinci Resolve 18 [11], a standard in the film restoration and video editing industry. The color correction intervention aimed at reestablishing the original tones of the films that were lost in the originals (Fig.2).

4.2 Sound materials

The sound materials were digitized by the CSC in 2024, following the methodology developed at the centre described in [12, 13], and refined in numerous digitization and restoration projects since [14].

The digitization process of the sound collection encountered several challenges due to the preservation conditions of the analog materials and, in some cases, poor sound quality, likely resulting from multiple transfers of the content over the years. In fact, on many occasions, Taroni used to work on open-reel tapes that were then copied on audiocassettes to take advantage of the possibilities offered by smaller portable formats and devices. For digitizing 1/4" open-reel tapes, three different open-reel recorders were used: the Studer A810 (up to 2-channel tapes), the Otari MX5050 (4-channel tapes) and the Fostex A-8 LR (8-channel tapes). Audiocassettes were digitized with a TEAC W-1200 double cassette deck. The A/D conversion was performed at 24 bit/96 kHz using Prism Sound Converters (*Lyra* and *Orpheus*). The digitization of the sound materials allowed us to identify the audiocassette containing the original audio loops used in the installation. For this installation, the artist created two electronic music compositions through the creative use of a phase modifier that intervened with the sound of a windshield wiper of a car, along with sounds produced by a sound effects generator and a Casio synthesizer. This combination of elements created two unique works, where the noise of the windshield wiper blends with synthesized melodies and sound effects. In the third composition, Taroni adopted a different approach, overlaying different contents on the two separate channels: the first channel features an excerpt from Orson Welles' famous radio broadcast *The War of the Worlds* (1938), while the second channel includes three swing-jazz songs (*Thinking of you* by Key Kayser, *When the moon comes over the mountain* by Kate Smith, and *Goodnight Sweetheart* by Bing Crosby). This overlay creates a fascinating contrast between the rhythm and the melody of the songs and the dramatic and eerie atmosphere of the radio excerpt, offering an engaging and multi-layered sound experience.

The audio tracks of *La(m)pelle di Ahmad* presented various issues, such as harmonic saturation, hum, and a high level of background noise. To address these issues and restore the audio to its original quality, the software iZotope RX 10 [15] was used for its advanced capabilities in noise reduction, spectral editing, and audio enhancement. Audio tracks were processed with De-Click and De-Crackle algorithms to remove impulsive noises, and then they were processed with a De-Hum filter to attenuate the low-frequency Hum. The background noise presented various profiles and intensities due to the fact that some of the noise was intro-

duced by the audiocassette carrier, while another component had been introduced by the original analog devices (open-reel tapes and phonographic discs) used for the first recording. Therefore, it was necessary to process the files acquiring multiple noise profiles while applying the De-Noise algorithm. The third composition was recorded on two separate channels of the audiocassette (Fig. 3), but during the live performances, the artist played it through a single speaker. Therefore, during the digital restoration process, the audio track was converted into mono. The whole restoration process took into consideration the interaction between the sound sources in the VR environment, especially in the final mixing and equalization process, in order to allow the user a better experience by removing resonances and unwanted energy on some frequencies.

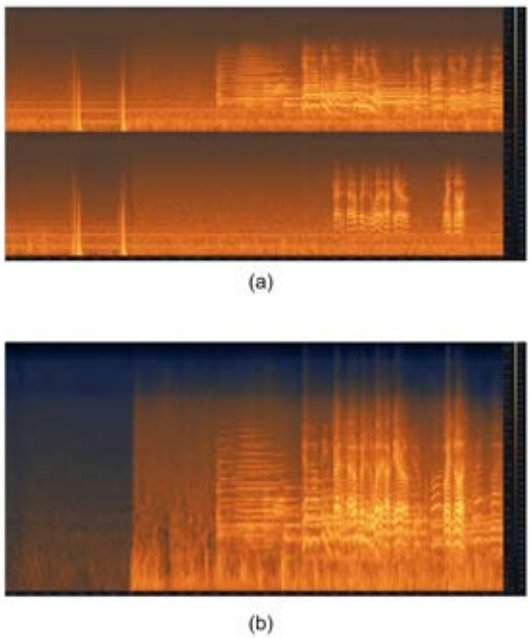


Figure 3. Original audio track before restoration (A) and after (B)

5. RE-ACTIVATION IN VIRTUAL REALITY

In re-activating the artwork, we considered the following requirements:

1. Accurate recreation of the physical features of the installation in the virtual environment;
2. Possibility to experience the artwork before and after the digital restoration process;
3. Accessibility of the artwork.

Given the relevance of the artwork's spatial aspect, we decided to recreate its typical real-life implementation in a 3D virtual environment, according to the artist's notes, instead of simply restoring and preserving the audio and films in digital form. Maximizing access to the artwork (requirement 3) meant having to work with dated technology so that the resulting application would be accessible to the widest range of VR headsets. We chose to work with two modern consumer VR headsets that are at the opposite ends of the technological state of the art: the Oculus Quest (2019), and the Meta Quest 3 (2023).

To meet these requirements, we needed a way to develop a VR application that was (i) flexible, to allow fast prototyping and iteration, and (ii) easily available, well documented, implementing open standards, and ideally (iii) open source, to ensure ease of future maintenance. We chose the open-source video game engine Godot [16]. First appearing in 2013, Godot is relatively new compared to the most popular engines commonly used for VR applications, such as Unity (2005) and Unreal Engine (1998). Despite this, Godot supports OpenXR, the open-source standard for virtual and augmented reality applications, and the documentation is of consistently good quality and frequently updated, making Godot a good choice for this project.

The virtual installation has been re-activated in two different versions: the first one includes the original digitized materials before any restoration intervention – i.e., faded film materials with magenta tones and noisy soundtracks – while the second one displays color-corrected films, accompanied by restored audio tracks. Pressing a controller's trigger allows the user to switch between the pre- and post-restoration versions.

La(m)pelle di Ahmad was typically implemented in a square or rectangular room, approximately 6×6 meters wide, and consists of three films projected on three of the four walls of the room, with three sound sources placed underneath the projection surfaces. All three projectors were Technicolor Instant Filmloop and were placed in the middle of the room, each facing its own wall. The projectors would be placed on a stand or, more frequently, hanging from the ceiling with wires.

5.1 Visual elements

We modeled the environment from scratch using Blender [17], based on the artist's instructions. We placed the projectors on a surface hanging from the ceiling using a rigid pole. This allowed us to limit the visual impact of the stand while at the same time avoiding issues with oscillating projectors, as they would be if hanging with wires. To not interfere with the artwork, we sought a balance between accuracy and clutter by only including the minimum visual elements to recreate the experience and excluding tangible items such as power cords, audio cables, unnecessary mounting parts, and so on. We did however include intangible analog elements such as the projectors' light flickering, because the artist felt it was an important part of the experience (Fig. 4). A 6-DoF visor enables the user to move inside the virtual environment to get around the projectors and closer to the projection screens. If the visor used does not support movement, we place the point of view near the empty wall, where the user can have a clear view of all the three projections.

Global Illumination (GI) was critical to achieve a minimum level of realism. Godot 4 does support several real-

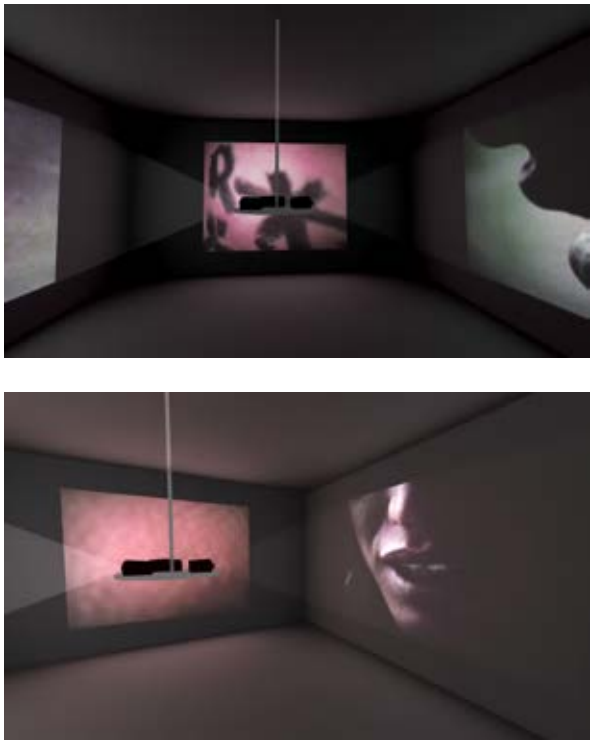


Figure 4. Vignettes from the re-activated VR installation.

time GI approximation techniques, such as Screen-Space Indirect Lighting (SSIL) and Signed Distance Field Global Illumination (SDFGI), but these are not available when deploying to stand-alone VR headsets, especially of the older generations. We resolved to fake GI by using regular lights, but found that the implementation was too computational demanding for the older headsets. The solution was to use Blender to bake the GI to a texture, and use it as an emissive texture in the scene, thus avoiding the expensive lighting implementation. By baking each light separately, we achieved the double objective of having high quality global illumination in the scene, and of being able to fake each projection's dynamic contribution by modulating each texture's colour and intensity, all for a very limited computational cost. While not entirely accurate, the result is convincing.

5.2 Aural elements

We placed positional audio sources in correspondence with the speakers. While Godot does not currently support binaural rendering, the illusion of spatialization is created through a combination of filters, effects, and panning, based on the relative positions of the sound sources, the listener, and the surrounding geometry. The audio sources in the original installation "interacted" with each other by raising and lowering their volume. Typically, when one source became louder, the other two became quieter. This behaviour was replicated in the VR implementation, introducing an additional level of uniqueness to each performance. Together with the 3 audio tracks, we included the motor noise of Super 8 projectors, which the artists indicated to be an important element of the original installation.

5.3 Evaluation

We performed a lightweight evaluation with a limited number of users during the iterative development process, looking at issues such as general discomfort, motion sickness [18, 19], and acceptability of the virtual audio-visual environment. During the time spent in the virtual environment, none of the testers reported any significant discomfort or motion sickness symptoms. Given the concept behind the original installation, the time spent by the user in the virtual environment was expected to last from 1 to 5 minutes. For such short sessions, the obtained results are then largely expected, and further formal testing will be performed once the re-activation is finalized and deployable on a larger scale.

In the near future, we intend to further evaluate the VR re-activation on more areas. The first one addresses usability, including issues of VR sickness and acceptance of the virtual environment, evaluating the re-activation with a larger number of users within a longer persistence period. The second is an evaluation of the installation's experience before and after digital restoration while also analyzing how user experience changes by including or removing analog elements such as projector noise and flickering. The whole evaluation will be accomplished by also providing users with archive documentation about the installation.

6. CONCLUSIONS

Virtual Reality has proven to be an interesting technology for re-activating and revitalizing multimedia art installations, especially for similar case studies. Through immersion in three-dimensional virtual environments, viewers can experience *La(m)pelle di Ahmad* in a completely new and engaging way, while still exploring the environment as it was conceived and created by the artist. Thanks to these technologies, the preservation intervention transforms into an immersive experience that helps in facing obsolescence-related problems typical of analog *Time-based media*. The use of VR improves access to artworks, allowing people with disabilities and users from distant locations to participate in the artistic experience without physically visiting the installation site. In fact, we plan to deploy the virtual experience on the artist's website where it will be freely accessible both as a traditional desktop-based 3D experience, and as a VR experience when using supported hardware. Furthermore, the virtualization of multimedia artworks such as *La(m)pelle di Ahmad* provides an innovative opportunity for users to experience the artwork before and after the digital restoration, both for audio and film components. Viewers can directly compare the differences and transformations brought about by the restoration process, enabling them to appreciate the technical aspects of the digital intervention and its impact on the artwork. This opportunity for direct comparison not only enhances the user experience but also contributes to critical understanding and artistic education, fostering greater awareness and appreciation for this particular art form.

During the entire lifespan of the project, from the digitization of the original materials through to the re-activation in a virtual environment, it has been of crucial importance working closely with the artist, and therefore having access to their knowledge of the artwork and the documentation collected during the artwork's original development and implementation. Without either of these, we would have found it much more difficult to make certain restoration and re-activation choices – e.g., selecting the correct audio tapes, and finalizing details in the virtual environment, such as the location and appearance of the original projectors, sound sources, and so on. If direct access to artists may become a challenge in time, we hope that this shows the importance of documenting the development of similar artworks, and the importance of preserving the related documentation as much as the artworks themselves. In the next future, we plan to further study the virtualization approach by applying it to other case studies of multimedia art installations, with the aim of defining guidelines within the MDP model, enabling more effective and targeted future virtualization interventions.

Acknowledgments

The authors would like to thank Roberto Taroni for providing the original audiovisual materials and for his enthusiasm and creative contribution in exploring new reactivation strategies, and Prof. Jennifer Malvezzi (University of Parma), whose early support made this project possible.

7. REFERENCES

[1] G. Youngblood, *Expanded Cinema: Fiftieth Anniversary Edition*. Fordham University Press, Mar. 2020.

[2] S. Canazza and G. D. Poli, "Four Decades of Music Research, Creation, and Education at Padua's Centro di Sonologia Computazionale," *Computer Music Journal*, vol. 43, no. 4, pp. 58–80, Dec. 2019.

[3] S. Canazza, G. De Poli, and A. Vidolin, "Gesture, Music and Computer: The Centro di Sonologia Computazionale at Padova University, a 50-Year History," *Sensors*, vol. 22, no. 9, p. 3465, May 2022.

[4] J. Malvezzi, F. Pitrolo, L. Cividin, and R. Taroni, Eds., *Taroni-Cividin: performance, video, expanded cinema (1977-1984) = edited by Jennifer Malvezzi, Flora Pitrolo*. Milano: Silvana, 2023.

[5] F. Bressan and S. Canazza, "The challenge of preserving interactive sound art: A multi-level approach," *International Journal of Arts and Technology*, vol. 7, no. 4, p. 294, 2014.

[6] A. Fiordelmondo, A. Russo, M. Pizzato, L. Zecchinato, and S. Canazza, "A multilevel dynamic model for documenting, reactivating and preserving interactive multimedia art," *Frontiers in Signal Processing*, vol. 3, p. 1183294, Jun. 2023.

[7] P. Gobira and A. Mozelli, "An Artistic and Curatorial Installation in Virtual Reality: The Development of an Artistic Low-Cost Interface at University," in *2016 IEEE International Symposium on Mixed and Augmented Reality (ISMAR-Adjunct)*. Merida: IEEE, Sep. 2016, pp. 209–214.

[8] A. Lockhart, "VR as a Preservation and Simulation Tool for Media Art Installations: ISEA 2020," Oct. 2020, pp. 700–703.

[9] H. Holling, "Virtualization as a Preservation Strategy for Computer-based Art: An (Empirical) Study into the Installation I/Eye (1993) by Bill Spinhoven van Oosten," 2011.

[10] R. Edmondson, "Audiovisual archiving: Philosophy and principles," United Nations Educational, Scientific and Cultural Organization, 2016.

[11] BlackMagicDesign, "DaVinci Resolve," 2023. [Online]. Available: https://www.blackmagicdesign.com/it/products/davinciresolve

[12] F. Bressan and S. Canazza, "A Systemic Approach to the Preservation of Audio Documents: Methodology and Software Tools," *Journal of Electrical and Computer Engineering*, vol. 2013, pp. 1–21, 2013.

[13] S. Verde, N. Pretto, S. Milani, and S. Canazza, "Stay True to the Sound of History: Philology, Phylogenetics and Information Engineering in Musicology," *Applied Sciences*, vol. 8, no. 2, p. 226, Feb. 2018.

[14] N. Pretto, A. Russo, F. Bressan, V. Burini, A. Roda, and S. Canazza, "Active preservation of analogue audio documents: A summary of the last seven years of digitization at CSC," in *Proceedings of the 17th Sound and Music Computing Conference 2020, SMC20*, Torino, Italy, 2020.

[15] Izotope, Inc, "Izotope RX," 2022. [Online]. Available: https://www.izotope.com/en/products/rx.html

[16] J. Linietsky, A. Manzur, and Various Contributors, "Godot Engine," 2024. [Online]. Available: https://godotengine.org/

[17] Blender Development Team, "Blender," 2023.

[18] J. Josupeit, "Cybersickness as the Virtual Reality Sickness Questionnaire (VRSQ) measures it!? – An environment-specific revision of the VRSQ," *Frontiers in Virtual Reality*, vol. 4, p. 1291078, Dec. 2023.

[19] H. K. Kim, J. Park, Y. Choi, and M. Choe, "Virtual reality sickness questionnaire (VRSQ): Motion sickness measurement index in a virtual reality environment," *Applied Ergonomics*, vol. 69, pp. 66–73, May 2018.

MUSIC THEATRE AND DIGITAL ARCHIVES

Filipa MAGALHÃES (anamagalhaes@fcsh.unl.pt) (0000-0003-4230-825X) [1]

[1] **CESEM FCSH, NOVA University of Lisbon**, Lisbon, Portugal

ABSTRACT

This article focuses on setting up a comprehensive digital repository (roadmap) capable of capturing the collective memory of music theatre, enabling sustainable access to performing arts works in Portugal and abroad, and thereby also enhancing their study, re-performance and visibility. This development spans three phases: proposing a terminological resource based on a literature review, existing knowledge organisation systems, and specialist inputs; selecting and producing ontological classes and properties for a case study, deploying recognised ontologies, such as Dublin Core, RiC-CM, and DOREMUS; developing a conceptual data model incorporating the semantic web and Linked Open Data principles to provide contextual information on music theatre composers and creators, including their biographies, work lists, and archive contacts. Interconnected by defined ontologies, this conceptual model provides users with a holistic view of these works. Representing music theatre works in digital archives challenges musicology, archival science, and digital humanities through requiring new approaches to ensure system interoperability. This digital archive helps to preserve unseen performing arts works while also strengthening Portuguese archives by interconnecting them with their international counterparts, enabling new scientific knowledge, historical narratives, and repertoire renewal able to reach wider audiences.

1. INTRODUCTION

I begin this article by briefly contextualizing the emergence of music theatre and thereby explaining the need to create a digital archive dedicated to the preservation of this performative genre. The appearance of music theatre lies at the origin of several intersecting events. Between 1955 and 1975, music theatre became a core focus of European composers as a result of the revolutionary experiences in musical language arising shortly after the end of World War II. Music theatre is associated with various composers, including Luciano Berio, Luigi Nono, Dieter Schnebel, Georges Aperghis, Sylvano Bussotti, György Ligeti, and Constança Capdeville, among others. In 1955, Berio and Bruno Maderna founded Italy's first electronic music studio, Studio di Fonologia Musicale, at RAI Milan. In this studio, composers were able to experiment with the interactions among acoustic instruments and electronically produced sounds as well as exploring new relationships between sounds and words [1].

Furthermore, composer John Cage's notions on the theatricalisation of electroacoustic music were key to establishing the foundations for music theatre. As Makoto Mikawa describes, in an interview with Roger Reynolds, Cage said that electroacoustic music needed to become theatrical through the introduction of live performance elements, people should do things and music should be visible [2]. These words profoundly impacted on Mauricio Kagel, whose first work already featured facets of theatricalization. The work is entitled *Antithese für einen Darsteller mit elektronischen und öffentlichen Klängen* (1962), and with the composer specifically portraying the piece as 'instrumental theatre'. In turn, Kagel's instrumental theatre influenced many of the aforementioned composers, all committed to various aspects of music theatre. Music theatre has since undergone multiple and significant transformations and helped to experimentally rethink theatrical traditions, artistic genres, performance conventions and the relationships between composers and society. This paradigm raises important questions regarding the relationship between music theatre and both earlier and more contemporary theories of drama, including: the usage of new technologies; the role of new venues and environments; the proposition of new conceptions for performers; and the challenges posed by music theatre for music analysis.

Music theatre works are inherently complex given their combination of the various artistic expressions specific to music, theatre, and dance with facets such as recordings, images, lighting, and scenery, all interacting as heterogeneous counterpoints [3]. Organising music theatre documentation thereby challenges traditional archival practices and requires new approaches. Music theatre sources are often scattered across institutional and personal archives, with some performative aspects missing entirely. Accordingly, to better understand the internal structures of music theatre works, it would be preferable to produce additional documentation. In some cases, only the testimonies from performers of past productions can provide the information necessary to completing particular aspects of the performance, but which also needs archiving. In addition, the representation of notation is idiosyncratic and non-conventional, with typically graphic scores including prescriptive indications rendered unclear by the lack of context. This requires evolving a taxonomy putting forward definitions for concepts still under debate in the performing arts. Most of the technological resources are already obso-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

lete and require digitisation with this process also requiring documentation for preservation purposes. All of these records must be incorporated into the archive to present these works comprehensively. The description of all related documentation improves our understanding of the structure of a work without, however, fixing it. Archivists, though experts in their field, often lack musicological terminology, hindering their ability to organise music theatre documentation effectively. Therefore, archives must include complementary documentation and list all relevant information. Organising music theatre works coherently thus requires collaboration between archivists and musicologists, addressing both terminological and methodological issues.

There still remains no publicly available digital archive interlinking the composers involved in music theatre productions either in Portugal or internationally. Consequently, this online platform aims to comprehensively preserve the collective memory of this performative genre of music theatre with a finalised digital archive representation. Developing this digital archive methodologically divides into three phases. A first phase proposing a terminological resource based on a literature review, existing knowledge organisation systems, and input from domain experts. Secondly, the process of selecting or designing ontological classes and properties, incorporating facets from recognised ontologies, for example from Dublin Core, RiC-CM [1] and DOREMUS. A third phase implies shaping the digital archive data model in keeping with the ontologies developed and applying the semantic web and Linked Open Data (LOD) [2] principles.

This archive will provide contextual information on music theatre works, including composer biographies, their works, and archive contacts. Each resource is to be described and interconnected to ensure a comprehensive view of these works, both in Portugal and internationally. This also aims to highlight the reality of Portuguese archives and position them side by side with their international peers, thereby exploring the practical implications of this archive, such as how it may serve educational purposes, support academic research and increase public engagement with the music theatre genre. This article showcases the platform's development through sketches and prototypes, focusing on the work *Double* (1982) by Constança Capdeville as a case study.

2. ONLINE PLATFORMS AS THE TOOL FOR PRESERVING MUSIC THEATRE HERITAGE

The design of this digital archive was inspired by the TKB platform that "allows for the construction of personal collections by the artists themselves, as well as the curation of various materials, which together create a network of relationships between the participating artists. The artists are able to import their materials and tag them according to their own idiosyncratic taxonomies, establishing ontologies and an interconnection between the various artists, thus expanding the network of connections between them" [4]. Within this same framework, Cristina Pattuelli et al. reflect on how building an ontology may support the evolution of knowledge representation on the Internet as well illustrated by the following quotation: "Ontologies can effectively serve as the knowledge backbone for integrating information, as their constructs, from class and property hierarchies to domain and range constraints, ground the mappings to heterogeneous data in a cohesive model. In general, the advantages of having our linked dataset defined through a sound and unified schema go beyond the task of supporting data integration. An ontology facilitates the parsing of the data, the automated processing and reasoning, and the detection of errors and inconsistencies" [5].

According to Alain Bonardi: "[t]he musicologist is at the same time a listener and a composer, since analysing a piece of music leads to 'rewriting' it" [6]. I regularly bear this idea in mind in keeping with how assembling the internal structure of any music theatre work does entail reshaping it even while retaining the relationships between the wide-ranging component parts. One simply cannot approach the internal structure of a music theatre work as one would a musical work inscribed in a score as the former incorporates various documents with dissimilar natures that interrelate as heterogeneous counterpoints. In music theatre works, "[t]here are traces of the composer's collaborative creative process in a wide variety of sources (scores, scripts, audio or video recordings, composer/musician sketches, notes, press reviews, images, slides, lighting directions, musical assistants) [...] and works are "characterized by the use of specific compositional processes and performance practices, and by collaborations with a multiplicity of authors, performers and technicians from different artistic fields. This complex process raises problems in terms of preservation" [7].

Thus, establishing a digital archive or roadmap capable of preserving music theatre involves three phases. The first relates to the terminological work, which involves reviewing the specialist literature, the knowledge organisation systems, and drawing on the contributions of experts in the field. The main outcome is a terminological resource that facilitates communication among performing arts experts and supports database construction. The second phase incorporates the development of the conceptual data model to establish access points that interlink different document types (e.g., scores, scripts, texts, audio/video recordings or images) to structure each work. This model captures the internal structure of works from a holistic perspective and is designed to assist archivists, composers, performers, choreographers, and other interested parties, while also facilitating research. Achieving this implies applying metadata schemas and web ontologies such as Dublin Core, RiC-CM and DOREMUS. The Dublin Core metadata standard focuses on simple and generic elements for describing resources, such as: 'Title' (creator or publisher), 'Author or Creator' (person or organisations responsible for the intellectual content), 'Description' (tex-

[1] More information available at: https://www.ica.org/en/records-in-contexts-conceptual-model (accessed on 22 February 2024).

[2] Further information available at: https://www.w3.org/DesignIssues/LinkedData.html (accessed on 5 March 2024).

tual description of the content), 'Date' (the resource available in its current form), 'Relation' (connections to other sources) and so forth. Certain archival sources reflect interconnected layers of past, present, and future contexts, with documents displaying dynamically accumulated orders. In turn, the RiC-CM model addresses these needs by improving archival descriptions and enriching them with cultural, social, and material contextual information. RiC-CM also overcomes the limitations of the classic hierarchical ISAD(G) [3] model, deploying a network structure to better manage the production, management, and safeguarding of documents. The DOREMUS ontology serves to describe and relate, for example, 'Events' (Expression Creation) or 'Works' (Complex or Individual). These tools help link data across the web using LOD and an interoperable framework. This conceptual model should also extend to other similar works that face similar preservation problems (e.g., installations). The third phase concerns the construction of the digital archive (roadmap), an interoperable online platform developed through Omeka, free and open-source software that serves as both a data management system and a web-publishing platform. By resorting to web ontologies, this platform will integrate resources from music theatre composers, mapping the relationships among them and connecting their key aesthetic approaches, practices, and methods (the arrows depict these links in Figure 2). The digital roadmap allows access to contextual information about music theatre including composer biographies, their respective lists of works/productions and contacts for both the institutional and personal archives containing these sources. Additionally, this helps preserve previously unseen performing arts works, enabling new scientific knowledge, historical narratives, and repertoire renewal, thereby reaching broader audiences.

There is below a representation of the digital platform through recourse to sketches and prototypes, adopting *Double* (1982), a music theatre work by Constança Capdeville, as a case study for structuring and interrelating digital archive documentation.

3. DIGITAL PLATFORM REPRESENTATION

3.1 Sketching

The digital archive (online platform) interface is then depicted on paper but only to communicate the core idea in keeping with the principles suggested by Bill Buxton [8] in his book *Sketching user experiences* (2007). The goal is to quickly sketch hand-drawn ideas that in no way represent the final solution but rather to communicate the earlier stages of an evolving idea.

The Figure 1 (on the left) broadly conveys the representation of music theatre composers as well as the relationships among them differentiated by colors; the second (on the right) depicts how, after clicking on Constança Capdeville, one accesses contextual information: her biography and list of works (still incomplete); then, in the third picture

[3] General International Standard Archival Description, for more information see: https://www.ica.org/resource/isadg-general-international-standard-archival-description-second-edition/ (accessed on 23 February 2024).

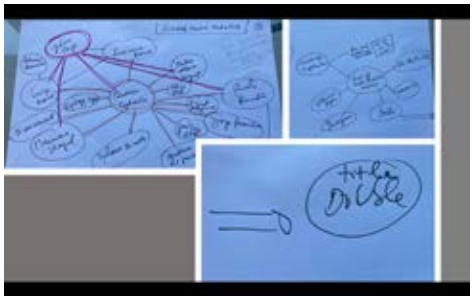


Figure 1. Sketching phase.

Figure 2. Prototyping: some music theatre composers.

(below right), by clicking on *Double*, there appears information on this music theatre work, which is not exemplified in this sketching but later features in the prototype.

3.2 Prototyping

The online platform is here represented in the form of a Prototype. Although the initial illustration may resemble the sketches, the prototyping displays a more conclusive shape with clearer and better organised results for presenting to users. This prototype is graphically represented in PowerPoint. [4] Figure 2 displays some of the composers linked with the music theatre genre and their respective relationships (with the arrow direction indicating who influenced who); while Figure 3 features contextual information on the composers and their works, including biographies and lists of works, in this specific case for the composer Constança Capdeville, with *Double* (1982) serving as a case study following its previous in-depth study in an archival context by the author [3] [7].

[4] Prototyping may contain interactions that are best portrayed by specific tools such as, for example, Figma (https://www.figma.com) or Justinmind (https://www.justinmind.com), among others.

Figure 3. Prototyping: Capdeville's biography, context and list of works.

By clicking on *Double* (1982) by Constança Capdeville, as indicated in Figure 3 (via the blue button with an arrow), a general description of the work appears. Figure 4 refers to a proposal by researcher Ana Caeiro, which intersects the ISAD (G) + RISM [5] archival standards. Caeiro's proposal seeks to combine both standards due to the limitations of the ISAD(G) guidelines for describing sacred repertoires and establishing points of access while retaining a perspective on musical works as a whole. Despite consisting of several sections, these form a composite document [9].

The fields featuring in the table (in Figure 4), despite the influence of Caeiro's proposal, align more closely with the ISAD (G) archival standard due to its better adaptation to the needs of music theatre works. In my proposal, the fields borrowed from the RISM guidelines are the 'Composer's name standardised', instead of 'Producer' [as suggested in the ISAD (G) archival standard]; the option for 'Date of composition', instead of 'Date of the manuscript', as it is easier to identify composition dates for contemporary music from the 1960s onwards; and finally, instead of 'Musical Incipit' (as put forward by the RISM guidelines), I opted for the incorporation of an audio or video excerpt. The purpose of the 'Musical Incipit' interrelates with the scope for this to be represented in and recovered from digital repositories by automated information retrieval systems. The 'Extent and medium of the unit of description' field, in turn, describes the list of materials relating to the work *Double*, in this case, mostly held by the National Library of Portugal (BNP).

However, as music theatre works combine different forms of artistic expression, embracing scenic-visual elements, recorded sounds, text, and so forth, and with some of these features serving as heterogeneous counterpoints, it becomes difficult to grasp their internal structure. In my opinion, the scope for visualising the work would perhaps therefore be more useful to understanding its structuring. This also opens up insights into how to curate information in music theatre works, given the nature of their relationships, and how such data information might best be stored in digital repositories. Caeiro's proposal, though relevant, does not meet the needs of music theatre works. Therefore, integrating various existing ontologies allows for the future aggregation of collected data and enriches archival descriptions. This requires new theoretical and practical approaches that traverse the fields of musicology, archival science and digital humanities.

Hence, to summarise, the table (Figure 4) describes the work *Double* as a compound document made up of several sections (simple documents). The 'Musical Incipit' is replaced by a video excerpt indicated as Intervention 2, but as the Prologue and Intervention 1 are analysed in this particular case, the link in the footnote corresponds to these two sections), given the complexity of portraying a work made up of heterogeneous sources, and not based only on a score as mentioned above. The 'Custodial and Archival History' field was added to facilitate the inclusion of relevant aspects about the work's conception and thereby describing the trajectory of the documentation in terms of its custody. For this type of work, it makes no sense to include a 'Uniform and Personal title(s)', or 'date of the manuscript' and 'date of composition' (as suggested in RISM) as, in contemporary music, this information is commonly inscribed in the score and/or programme notes (and thus precisely known), therefore the work's title and date was included; the standardised composer's name corresponds to the producer, who, in this particular case, was Constança Capdeville herself [9]. The 'Audio/video excerpt' field, referenced in the same table (see Footnote) represents the Prologue and Intervention 1 of the work *Double* (1982) by Constança Capdeville. [6] This representation provides a sort of listening guide, then produced by iAnalyse, [7] a software developed by Pierre Couprie.

Reference code(s)	PT/BNP/CC/ [...]
Level of description	Compound document
Title of work	Double
Composer's name standardised	CAPDEVILLE, Constança
Date of work's creation	1982
Extent and medium of the unit of description	1 folder • Original full score (297 x 420 mm): Paper • Photocopies of scores with and without annotations (297 x 420 mm): Paper • Full score (297 x 420 mm; scenic show) • Outline for handwritten programme note (210 mm x 297): Paper • 6th Gulbenkian Contemporary Music Meetings" Programme (148 x 210 mm): Paper • Photographs (27,6 x 19,5 cm): Film material • Script (210 mm x 297): Paper • Script(s) for sound, light and more show (210 mm x 297): Paper • Composer's notes (210 mm x 297): Paper 1 box • 2 original magnetic tape Double mono, 1/4 inch tape width, recording speed: 7 ½ ips (19 cm/s); PET
Scope and content	Music collection related to Constança Capdeville's professional activity as a composer. Includes original handwritten scores, full scores, programme of the performance, composer's notes, script(s), press clippings, photographs, audio recordings (magnetic tape).
Custodial and Archival History	The work *Double* was conceived by Constança Capdeville, commissioned by the FCG within the scope of the 6th Gulbenkian Contemporary Music Meetings. The work was premiered on June 5, 1982, at the Auditorium II of the FCG under the direction of the composer herself, the only performance of the work to date. *Double* was composed for voice, cello, piano, two percussionists (who play chess on stage), music clock, two magnetic tapes and lights. *Double* is part of the music theatre composer's production, and it is divided into ten sections: 1 prologue, 8 interventions (1, 2, 3 [...] 8) and 1 epilogue. On the premiere date, the performers were António Wagner Dias (baritone), João Paulo Santos (piano), Luísa Vasconcelos (cello), Jorge Primitho (chess player - percussion), João Heitor (chess player - percussion), Carlos Alberto Augusto (recorders operator), 7 dancers from Dança Grupo (music clock). After Constança Capdeville's death in 1992, her entire collection (including documentation referring to *Double*) was kept in the house of Janine Moura (a personal friend of the composer), who created a preliminary record; in 2012 the collection of Constança Capdeville was donated to the BNP by Manuel Cintra (at the time the widower and official heir of Capdeville); the documentary estate was delivered to BNP in September 2015.
Audio/video excerpt	https://drive.google.com/file/d/1xRUZ5SQNsTWqsudaTK0NEEC2jbRuHMcB/view?usp=sharing (Intervenção 2 correspondente à obra *Double*)
Notes	At the Gulbenkian Foundation Archive there is an audio recording of *Double*'s 1982 debut. Other materials such as scores with specific notes for the performers are in their own possession, integrating their personal archives, namely: João Paulo Santos (piano), António Wagner Dias (voice) and Carlos Alberto Augusto (prescriptive score for magnetic tape recordings); the dancer Luísa Vendrell has in her possession a photograph of Dança Grupo members (as it was composed at the time).

Figure 4. Table containing an archival description of the work *Double* deploying the ISAD(G) + RISM archival standards with adaptions for contemporary music.

4. UNDERSTANDING THE INTERNAL STRUCTURE OF MUSIC THEATRE WORKS

This section emphasises RiC-CM as this conceptual model overcomes the limitations of the classic hierarchical standards for archival description, ISAD(G), ISAAR, [8] ISDF [9]

[5] A specific guidance for the description of primary musical sources, see: https://rism.info (accessed on 22 February 2024).

[6] https://drive.google.com/file/d/1xu2suD23KNgaLTLdZukY7NNfFu2wGxy6/view

[7] For further information, see: https://logiciels.pierrecouprie.fr (accessed on 23 February 2024).

[8] ISAAR (CPF): International Standard Archival Authority Record for Corporate Bodies, Persons and Families, 2nd Edition, see: https://www.ica.org/resource/isaar-cpf-international-standard-archival-authority-record-for-corporate-bodies-persons-and-families-2nd-edition/ (accessed on 23 February 2024).

[9] ISDF: International Standard for Describing Functions, see: https://www.ica.org/resource/isdf-international-standard-for-describing-functions/ (accessed on 23 February 2024).

Figure 5. RiC – CM integrating current ICA description standards.

and ISDIAH, [10] by incorporating a network structure to better manage the production, handling, and safeguarding of documents. Thus, this consists of depicting a schematic representation based on the RiC-CM conceptual model, describing the internal structure of a music theatre work in conjunction with the relationships prevailing among the various components and audio and video notes/excerpts, which correspond to additional documentation. This builds up a holistic perspective on the respective work while nevertheless maintaining interoperability between information systems. As described in the RiC-CM manual published by the International Council on Archives (ICA), this conceptual model is "intended to serve as a foundation for describing records to facilitate their near- and long-term preservation and use. It provides a conceptual framework based on archival principles for designing and implementing standardised systems for the intellectual control and description of analogue and digital resources in records management and archival programs, including description of the contexts in which the resources originated and were used, as well as the contexts of ongoing subsequent management and use". [11]

Furthermore, and also according to the ICA, through its reconciliation and integration, the RiC conceptual model is designed to overcome the limitations of the existing archival standards, the aforementioned ISAD(G), ISAAR, ISDF and ISDIAH, precisely because there is still no common and consistent archival description model able to reflect the contextual complexity of documents, and particularly important for collections in the performing arts (see Figure 5). The ICA also issued the Records in Contexts-Ontology (RiC-O), an OWL ontology to describe archival record resources and their contextual entities.

4.1 Case study: *Double* (1982) by Constanca Capdeville

Double (1982), by Constança Capdeville, was selected as the case study given the internal structure of this music theatre work had already been extensively studied as part of the author's doctoral research. In turn, *Double* presents numerous challenges to its archiving and preservation. This work is divided into ten parts [Prologue, Intervention 1, 2 (...) and 8, Epilogue] but, in this specific case, I focus only on the Prologue and Intervention 1. [12] Most of the materials belonging to *Double* are currently stored by the BNP, although there is also a tape recording of the 1982 premiere stored in the Calouste Gulbenkian Foundation Archives (named as FCG tape). Some graphic and prescriptive scores, containing specific indications for the performers, remain the personal property of the respective performers. For example, pianist João Paulo Santos holds a score with the piano section he performed at the premiere. Furthermore, Carlos Alberto Augusto, responsible for editing and operating the tape recordings during the premiere, retains his own score detailing the specific indications given to him by Capdeville.

As mentioned above, prior analysis of the internal structure of *Double* took place through recourse to the following documents: "1) scores from Capdeville's collection and those belonging to the performers containing their specific instructions; 2) scripts for the sound, lighting, and mute choir; 3) magnetic tapes 5 and 6, both used in the original performance; 4) the composer's notes, excerpts from texts, programme note, images, and other documents; 5) a recording of the live performance from 1982 (hereafter: the FCG tape). Even following access to these materials, I still encountered various difficulties in retrieving widely dispersed documents; understanding the various and diverse layers of information they contain, including the distinction between acoustic and electronic sounds; and ascertaining what the gestures and movements were due to the lack of information, especially for the mute choir." [3] The mute choir was made up of seven dancers from Dança Grupo.

The systematisation of the existing information on the structure of *Double* covered the following: "1) to retrieve documents and audio sources and then transfer them to current formats; 2) to analyse the score in relation to the recording of the live concert, resolving inconsistencies by referring to the sound, light, or mute choir scripts, themselves often dispersed and incongruous; 3) to study the two tape recordings (tapes 5 and 6), identifying each sound to understand the tape inputs throughout the live performance; 4) to interview performers involved in the original 1982 performance. The aural structure of *Double* was reconstructed from the recordings (tapes 5, 6 and FCG tape), and by comparing documents. As a preliminary step, I analysed the waveforms of the sounds, particularly those moments when live sounds were mixed with recorded sounds." [3].

Figure 6 below describes in detail the beginning of *Double* (1982), thus, the Prologue and Intervention 1 as both are interconnected. To explain the conceptual model, I analysed the original score [13] from pages 1 to 4, which mutually interrelates several different types of documents (au-

[10] ISDIAH: International Standard for Describing Institutions with Archival Holdings, see: https://www.ica.org/resource/isdiah-international-standard-for-describing-institutions-with-archival-holdings/ (accessed on 23 February 2024).

[11] Source: International Council on Archives Expert Group on Archival Description, Version 1.0 November 2023, p. 5. Available at: https://www.ica.org/app/uploads/2023/12/RiC-CM-1.0.pdf (accessed on 24 February 2024).

[12] The following link leads to the project developed to organise the internal structure of the work: `https://gitmind.com/app/docs/ml42ccc6` (accessed on 20 February 2024). GitMind software helps user visualise and organise their ideas, projects, and workflows.

[13] I refer to the original score because, as described in Figure 4, in the BNP collection belonging to Capdeville, in addition to the original score, there are several versions of scores and copies similar to the original one, with and without notes, the ideal would be to gather all the additional notes in a single document to ensure the most complete and integrated information.

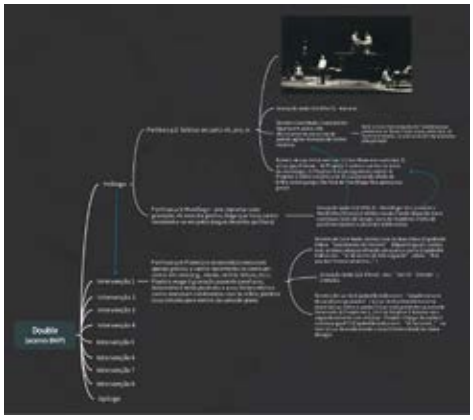


Figure 6. Describing the interactions between the *Double* (1982) documents: Prologue and Intervention 1.

dio recording script, lighting script, mute choir script, and additional documentation). The mute choir script sets out gestural instructions for the dancers who only performed gestures and did not produce any sounds that were all performed by audio recordings. The photograph included in Figure 6 is the only one extant of the original performance and particularly important because, given there is no video of this performance, we at least gain an idea of how the performers, instruments and other objects were located on stage. Below, there is a brief explanation of the interactions occurring across the different documentation. In the Prologue, at 'Partitura [14] p. 2' the soloists on stage and related documents are indicated, for example, in the mute choir script, the audio recording S1A (Tape 5) and the lighting script; then, in 'Partitura p. 3' only the S1B (Tape 5) audio recording is indicated, as it is a different sound, and all other documents are equal for both score pages (with arrows indicating this association). The same connection arises between the Prologue and Intervention 1, with both sections amalgamated with this logic applied to interrelating the information provided by the several documents. Subsequently, there is a description of the contents of each document, such as: image, sounds, visible notes and links.

Double (sources from the BNP archives)

Prologue Score p. 2: Soloists on stage: cello, piano, voice

- Image of performers on stage
- Audio recording S1A (Tape 5) – Maracas `https://drive.google.com/file/d/14Um9BznUtzxd3ExhGyFyv7sEzfMk3xDp/view?usp=share_link`
- Mute Choir Script: Crossing figures on stage; go discreetly behind the panel; shake maracas violently.
- Note: The Choir is a group of 7 dancers who belonged to the Dança Grupo and who, in this work, make only movements with their voices reproduced by the recording.
- Lighting Script: Beginning without light 1) Mute Choir in backlight 2) a) Tenuous general light; - b) Projector 1 over the singer at the beginning of the monologue. 3) Projector 4 on chess players 4) Projector 2 on a shelf with 5) Light generating a glow effect on the gong (at the end of the monologue there is only general light) Score p.3: Monologue - piano reproduces recording sounds; cello performs gestures, pretends to touch; singer moves on stage (follows sheet music drawing)
- Audio recording S1B (Tape 5) - Monologue: Voice reciting the Manifesto (Picasso) and vocal effects; Tuning fork bottom (continuous sound); Gong sound; Wood sounds; Screw effect (piano) and pizzicato (cello) `https://drive.google.com/file/d/16ZHgWAMorj2YK4HGQZkxrsNvLaw8Z5aG/view?usp=share_link`

Intervention 1 Score p.4: Pianist and cellist perform only gestures; the singer moves like singers on stage (e.g., rehearsal, recital, reading, etc.); Pianist reacts to the recording by pulling screws; Cellist mimes pizzicato and bow; Cellist and singer perform movements with their hands; pianist plays crotales inside the piano's resonance box

- Mute Choir Script: Gestures with both hands (Capdeville indicates: "birth of man" (Michelangelo); Gestures with both heads looking at each other (Capdeville indicates with: "question mark look"; cue: "But you don't know what it is...")
- Audio recording S2A (Tape 6) - Voice: "Voi Ch'Entrate"; Crotales `https://drive.google.com/file/d/1ZoBd34cOTMV5xO19M9-B2xsvT1AYSLMZ/view?usp=share_link`
- Lighting script: 6) (Capdeville indicates with: "sound sequence of engraved screws") a) Light most present in the piano area b) Light on the panel c) Light most present in the cello area d) Projectors 1, 2 and 3 (Projector 3 illuminates a second shelf with music) - Projector 4 (chess game) remains the same 7) (Capdeville indicates with: "et facta est..." in his voice) a) Golden light invades the scene b) Intensity light dims slowly.

5. RESULTS

This section presents the results of a survey carried out with two colleagues, who were shown sketches and prototypes designed to represent the digital platform in order to obtain their feedback. This task took place in the context of the curricular unit 'Introduction to Virtual Environments and Web Interfaces', part of the Master's Degree on Digital Curation and Digital Humanities run by the Faculty of Social Sciences and Humanities, NOVA University of Lisbon, Portugal, on 4 January 2024, taught by Professor Armanda Rodrigues. Colleague 1 felt the idea remained

[14] The Portuguese word for score.

somewhat confusing for someone who did not know the respective scientific domain, in this particular case, contemporary music and archives. Colleague 1 stated that users would certainly not be someone from the field of literature as was their particular case. According to this colleague, a future user must understand the terminology applied in this scientific domain and that required a particular interest in the subject (in practice, the target community for this platform are researchers, archivists, composers, performers, choreographers and other interested parties). Colleague 1 considered one of the limits of the project to be the fact it is very specific. Graphically speaking, Colleague 1 also reflected that the biggest problem stems from the excessive processing of data necessary for adaptation to an eventual platform. The colleague considers it an excellent idea but a monumental and overly ambitious project. For the purposes of the prototype, Colleague 1 noted that it should stick to representing only one part of the work as an example. After questioning this overly ambitious scope of the project, Professor Armanda Rodrigues intervened to say that it would be interesting to understand how these components are mapped and represented digitally. In fact, this project is still a work in progress. After explaining the project concept to Colleague 2 when still in the sketching phase, the latter stated that it lacked something more interactive. Colleague 2 also felt that this was a very specific area, and he did not quite know how he could be of assistance. However, he added it might make sense to build a vertical (in-depth) prototype, as he considered this model more interesting, and that musical components should be inserted, such as audio, video and images, as well as additional notes, in order to facilitate visualisation and emphasise the relationships between the various component parts. In fact, as this constituted the initial idea, this was incorporated into the final prototype design.

6. CONCLUSION

To conclude, a digital repository capable of envisioning the previously hidden collective memory of music theatre composers will provide sustainable digital access to previously unknown music theatre works, enhance historically informed performances; undertake terminological work in the performing arts; and organise the knowledge necessary to designing this digital roadmap. This will benefit researchers, archivists, performers, composers, stage directors, choreographers and other interested parties while also serving educational purposes, supporting scholarly research, and deepening public engagement with the music theatre genre. Additionally, this digital platform seeks to strengthen the Portuguese archival landscape in the performing arts and establish links with international archives for mutually beneficial exchanges. Moreover, the digital archive serves as both a relational and interoperable database and a website display platform, highlighting overlooked composers in the performing arts and boosting the profile of their works. Thus, this represents a valuable resource for society in general.

Acknowledgments

This work is supported by national funding from the FCT – Fundação para a Ciência e a Tecnologia, I.P., under the Scientific Employment Stimulus—Individual Call—[CEEC Individual 2022–2022.07416.CEECIND: https://doi.org/10.54499/2022.07416.CEECIND/CP1725/CT0047]. It also receives support from CESEM – Centre for the Study of the Sociology and Aesthetics of Music [UIDB/00693/2020: https://doi.org/10.54499/UIDB/00693/2020] and IN2PAST – Associate Laboratory for Research and Innovation in Heritage, Arts, Sustainability and Territory [LA/P/0132/2020: https://doi.org/10.54499/LA/P/0132/2020].

7. REFERENCES

[1] F. Magalhães, "A obra de Constança Capdeville: itinerários artísticos, sociais e afetivos," in *Geografias Culturais da Música, do Som e do Silêncio.* Braga: Universidade do Minho (Lab2PT), 2021, pp. 275–99.

[2] M. Mikawa, "The theatricalisation of Mauricio Kagel's "Antithese" (1962) and its development in collaboration with Alfred Feussner," 2021.

[3] F. Magalhães, "Musicological Archaeology and Constança Capdeville," *TDR: The Drama Review*, vol. 66, no. 3, pp. 64–77, Sep. 2022. [Online]. Available: https://www.cambridge.org/core/product/identifier/S1054204322000302/type/journal_article

[4] F. Magalhães, Filipa, "Idiosyncratic Ways of Preserving Performing Arts Creation in an (Digital) Archive," in *Proceedings of the International Conference on Technologies for Music Notation and Representation*. Marseille: Les éditions de PRISM, 2022.

[5] M. C. Pattuelli, A. Provo, and H. Thorsen, "Ontology Building for Linked Open Data: A Pragmatic Perspective," *Journal of Library Metadata*, vol. 15, no. 3-4, pp. 265–294, Oct. 2015. [Online]. Available: http://www.tandfonline.com/doi/full/10.1080/19386389.2015.1099979

[6] A. Bonardi, "IR for Contemporary Music: What the Musicologist Needs," in *Proceedings of the First Annual International Symposium on Music Information Retrieval (ISMIR 2000)*, 2000.

[7] F. Magalhães, "Material Sources, Lack of Notation and the Presence of Collaborators: The case of *Double* by Constança Capdeville," *Organised Sound*, vol. 27, no. 2, pp. 193–205, Aug. 2022. [Online]. Available: https://www.cambridge.org/core/product/identifier/S1355771822000346/type/journal_article

[8] W. Buxton, *Sketching user experiences: getting the design right and the right design.* Amsterdam Boston: Elsevier/Morgan Kaufmann, 2007.

[9] F. Magalhaes, "Constança Capdeville's Personal Archive: Difficulties in Describing and Documenting Performative Practices," *Archival Notes (Source Studies in Twentieth- and Twenty-First-Century Music)*, no. 7, Dec. 2022.

PHASE REPAIR FOR TIME-DOMAIN CONVOLUTIONAL NEURAL NETWORKS IN MUSIC SUPER-RESOLUTION

Yenan ZHANG[1], **Guilly KOLKMAN**[2], and **Hiroshi WATANABE**[1]

[1]*Graduate School of Fundamental Science and Engineering*, **Waseda University**, Tokyo, Japan
[2]*The Faculty of Science*, **University of Amsterdam**, Amsterdam, Netherlands

ABSTRACT

Audio Super-Resolution (SR) is an important topic as low-resolution recordings are ubiquitous in daily life. In this paper, we explore the music SR task through solo piano music, which is challenging due to the wide frequency response and dynamic range of music. Many SR models exploit Time-Domain Convolutional Neural Network (TD-CNN), which benefit from the joint processing of magnitude and phase information of audio signals. However, prior works indicate that TD-CNN approaches tend to produce annoying artifacts, and the cause of the artifacts is yet to be identified. In this paper, we demonstrate that the artifacts in TD-CNNs are caused by the phase distortion via a subjective experiment for the first time. We further propose Time-Domain Phase Repair (TD-PR), which uses a neural vocoder pretrained on the wide-band data to repair the phase components in the waveform outputs of TD-CNNs. The proposed TD-PR obtained better mean opinion score than TD-CNN baselines, which demonstrates TD-PR significantly improves the perceptual quality of TD-CNNs. Since the proposed TD-PR only repairs the phase components of the waveforms, the improved perceptual quality in turn indicates that phase distortion has been the cause of the annoying artifacts of TD-CNNs. Moreover, the proposed TD-PR can be easy applied to arbitrary TD-CNNs without additional adaptation. Audio samples are available on the demo page [1] .

1. INTRODUCTION

Audio Super-Resolution (SR), also known as bandwidth extension and bandwidth expansion, aims to predict the High-Resolution (HR) components from the Low-Resolution (LR) input audio. Audio SR is an important topic as LR audio is common in daily life, *e.g.*, historical recordings or unprofessional-made modern recordings. As the real-world LR recordings have a variety of bandwidths, addressing audio SR in real world is challenging. In recent years, Deep Neural Networks (DNNs) have become the mainstream in audio SR tasks [1–4], but only a few works focus on the music [2]. In this paper, we focus on solo piano recordings as a representative to investigate the music SR task.

Various works have delved into the DNN-based approaches for audio SR. Frequency-Domain Convolutional Neural Networks (FD-CNNs) aim to directly recover the HR components in the magnitude spectrogram, and generally require additional signal processing to estimate the corresponding phase information, such as Griffin-Lim algorithms [2] or neural vocoders [4]. Compared with FD-CNN methods, Time-Domain Convolutional Neural Networks (TD-CNNs) that directly learn a wave-to-wave mapping, are considered being able to avoid the phase problem in audio SR tasks [2]. However, TD-CNNs tend to produce annoying artifacts in their waveform output. To alleviate the artifacts, Lim *et al.* proposed a time-frequency hybrid model [5] based on AudioUNet. Wang *et al.* made efforts on objective function that employing the frequency domain losses [6] during the TD-CNN's training. The data augmentation strategy was proposed in [7] to improve the robustness of TD-CNNs.

Although the above efforts for TD-CNNs improved audio SR quality measured by objective metrics, none of the above TD-CNN methods succeeds in removing the artifacts according to their open-available audio samples. We hypothesize that the inconsistency between objective and subjective evaluation results could have been caused by some signal components that cannot be measured by the objective metrics. We observe that phase components are not explicitly measured by typical objective metrics such as log-spectral distance. This observation encourages us to explore the importance of phase in audio SR tasks. In terms of up-sampling ratio, many works perform the SR on a fixed ratio (*e.g.*, $2\times$) [1, 2], which would be a limitation when apply these models to real world scenarios.

We investigate the artifacts of TD-CNNs in the following ways. First, we train three TD-CNNs with different architecture and parameter amount to handle LR music with various bandwidth, which is applicable to real world problems. The SR capability of three TD-CNN baselines as well as the artifacts are successfully reproduced. Second, we conduct an AB listening test which, to the best of our knowledge, is the first to demonstrate the artifacts in TD-CNNs are caused by the phase distortion via a subjective experiment. Last but not least, we propose the Time-Domain Phase Repair (TD-PR) method, which utilizes a vocoder pretrained on wide-band music signals to repair the distorted phase components in the waveform output of the TD-CNN. Since the vocoder and TD-CNNs are trained

[1] https://mannmaruko.github.io/demopage/tdpr.html

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

independently, a single pretrained vocoder can be directly applied to arbitrary TD-CNNs without additional adaptation. Therefore, we apply TD-PR to the aforementioned three TD-CNNs. The proposed TD-PR consistently and significantly improved the perceptual quality of all three TD-CNN baselines. Since TD-PR only repair the phase components of the waveforms, the improved perceptual quality in turn indicates that phase distortion has been the cause of the annoying artifacts of TD-CNNs.

2. RELATED WORK

Various approaches for audio SR have been developed and some of them work in Frequency Domain (FD). Li *et al.* proposed an FD approach for speech SR, which consists of 2 steps [8]. The first step is mapping the magnitude components from narrow-bandwidth to wide-bandwidth by DNN. The second step is to estimate the corresponding phase by signal processing. Following this work, Hu *et al.* introduced Generative Adversarial Network (GAN) into both steps and got the better performance [2]. However, training two GAN-based models is difficult due to the instability of GAN training. Furthermore, this SR system works on a fixed up-sampling ratio, which limits its application to real world problems. Liu *et al.* used a GAN-based neural vocoder for the second step without using GAN in the first step, which successfully performed speech SR with the ability of handling various up-sampling ratios [4]. It is worth pointing out that the FD approaches mentioned above requires strict matching of mel-spectrogram settings between the FD-CNN model and the neural vocoder. Therefore, some FD-CNN models trained with an unmatched mel-spectrogram settings cannot directly work with the pretrained vocoder.

Contrary to FD approaches, TD-CNNs are considered being able to avoid the phase problem in audio SR tasks due to the direct waveform processing [2]. AudioUNet is one of the pioneers of tackling audio SR by a TD-CNN [1]. Tagliasacchi *et al.* proposed SEANet [9], a GAN-based model for speech SR. The generator of SEANet is a light-weight but effective TD-CNN. In this paper, we utilize the generator of SEANet to music SR as one of our baselines. Defossez *et al.* proposed a TD-CNN model referred to as Demucs, which is a large model with over 130M parameters and is initially designed to address music source separation [10]. Considering the fact that Demucs has shown strong performance in tasks besides source separation [11], we utilize the Demucs model in the SR task in this paper. To the best of our knowledge, this is the first time to apply Demucs to the music SR task.

The mel-to-wave transform is commonly addressed by neural vocoders. TFGAN is a light-weight vocoder [12] and has been applied to the speech SR task [4].

3. PROPOSED METHOD

3.1 Time-Domain Phase Repair

In order to alleviate the artifacts caused by distorted phase components, we propose Time-Domain Phase Repair (TD-PR). The TD-PR framework consists of two separately pretrained DNN modules and a phase replacement operation.

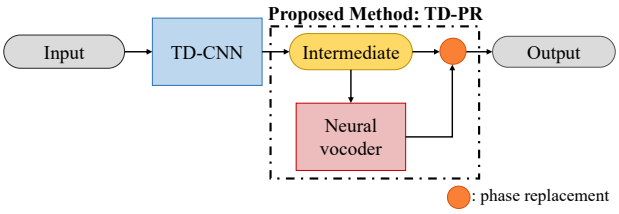


Figure 1. Overview of the proposed TD-PR: The TD-CNN is trained to perform super-resolution for various narrow-band inputs. The neural vocoder takes only the magnitude of the TD-CNN's output as input, and re-synthesizes another waveform that contains repaired phase components. Then, the distorted phase components in TD-CNN's output are replaced by that from the vocoder.

The overview of the proposed method is shown in Fig. 1. Specifically, the TD-PR pipeline involves the following steps. First, a TD-CNN is trained to perform muisc SR. To handle LR music with various bandwidths which is common in real world, we apply a simulation pipeline to HR music data to get the corresponding LR version. With the simulated pseudo paired data, the training of TD-CNN for music SR is made possible. Details of the simulation pipeline and training objectives are explained in the succeeding section.

Second, we pretrain a neural vocoder on the unprocessed HR music data. Since a neural vocoder can generate realistic waveform signals with only the magnitude input, it can be inferred that a vocoder can generate realistic phase components that are coherent with the input magnitude components. This inspires us to utilize a neural vocoder to repair distorted phase.

Last, we introduce TD-PR to repair the phase components of the output from the TD-CNN. The intermediate waveform produced by the TD-CNN is decomposed into magnitude and phase components by Short-Time Fourier Transform (STFT). We empirically use an STFT of 1024-point hann window and 256 hop length for a sampling rate of 16 kHz. The neural vocoder takes only the magnitude of the TD-CNN's output as input, and re-synthesizes another waveform that contains repaired phase components. Then, the distorted phase components in TD-CNN's output is replaced by that from the vocoder, and a phase-repaired waveform output is produced by inverse STFT. Although the vocoder also outputs waveform, we decide not to use it as the final results, because empirically we found that the vocoder could introduce distortions in the lower frequency part.

According to the above description, the vocoder and TD-CNNs are trained independently, which indicates a single pretrained vocoder can be directly applied to arbitrary TD-CNNs without additional adaptation, making the method flexible. It is worth noting that since TD-PR only repair the phase components of the waveforms, the improved perceptual quality in turn indicates that phase distortion has been the cause of the annoying artifacts of TD-CNNs.

3.2 Simulation Pipeline

The design of simulation pipeline has been shown important to the performance and robustness of audio SR models [6, 7]. The simulation pipeline we utilize mainly follows the principles in [6, 7]. Specifically, we simulate each LR input by randomly choosing a low-pass filter from 7 low-pass filters, including Butterworth, Chebyshev type 1, Chebyshev type 2, Elliptic, Bessel, subsampling (*i.e.*, resample_poly in scipy), STFT filter (*i.e.*, replacing the high frequency components with zero elements) with the filter order randomly selected from 6 to 10. We use the implementation of low-pass filters provided by Liu *et al.* [2] [4].

Since 3 kHz has been analyzed to be the typical bandwidth of real historical recordings [13], we sample an LR bandwidth between 2.5 kHz and 4 kHz via a uniform distribution. We don't consider 2 kHz because we found this bandwidth will filter out a part of melody, which is not common in real recordings. The low-pass filtering is conducted on-the-fly during training.

3.3 Loss Function

Inspired by [6], we perform cross-domain loss to guide TD-CNNs to capture features in both time and frequency domains. The loss function (denoted as L) is comprised of two parts, multi-resolution STFT loss (L_{MRSTFT}) [14] and multi-resolution wave loss(L_{MRwave}) which is similar to L_{MRSTFT}. The loss function is defined as below:

$$L = L_{\text{MRSTFT}} + \lambda L_{\text{MRwave}}, \tag{1}$$

where λ denotes the hyperparameter balancing the two loss terms. In our case, we empirically set $\lambda = 1000$ to balance the weights between two losses.

The definition of L_{MRSTFT} and L_{MRwave} are shown as follows:

$$L_{\text{MRSTFT}} = \frac{1}{M}\sum_{m=1}^{M} L_{\text{STFT}}^{(m)}(y, \hat{y}), \tag{2}$$

$$L_{\text{MRwave}} = \frac{1}{N}\sum_{n=1}^{N} L_{\text{wave}}^{(n)}(y, \hat{y}), \tag{3}$$

where y and $\hat{y}$ denote the ground truth and generated sample respectively. M denotes the number of STFT losses with different analysis parameters (*i.e.*, FFT size = [512, 1024, 2048]; hop size = [256, 512, 1024]; window size = [512, 1024, 2048]). We use the implementation of L_{MRSTFT} from [15]. N denotes the number of wave losses with different sampling rate (*i.e.*, original sampling rate, $2\times$ down sampling rate, $4\times$ down sampling rate).

L_{wave} is defined as follows:

$$L_{\text{wave}}(y, \hat{y}) = \frac{1}{P}\| y - \hat{y} \|_1, \tag{4}$$

where P denotes the number of wave samples and $\| \cdot \|_1$ denotes the L1 norms.

[2] https://github.com/haoheliu/ssr_eval

4. EXPERIMENTS

4.1 Dataset and Implementation

We trained and evaluated our model on the MAESTRO dataset [16]. It is composed of about 200 hours of high-quality classical piano recordings in waveform. Although these recordings have the sampling rate of 44.1 kHz or 48 kHz, we empirically found that 16 kHz is high enough for the piano solo. Hence, we performed music SR with the target bandwidth of 8 kHz, *i.e.*, a target sampling rate 16 kHz. We used the official split of the MAESTRO dataset for training, validation and test. We cut all of the waveform into 30-second short clips for efficient training.

To implement the proposed TD-PR framework, we trained a TFGAN [12] from scratch on MAESTRO training set by using an unofficial implementation [3]. We followed the original settings, except resetting the sampling rate to 16 kHz, and trained it for 1M iterations.

Since TD-PR is feasible for arbitrary TD-CNNs with a single pretrained neural vocoder as mentioned in Sec. 3.1, we evaluated TD-PR with three representative TD-CNN models as baselines: AudioUNet [1], Demucs [10] and SEANet generator [9]. We trained them from scratch with the loss function mentioned in Sec. 3.3 by applying the simulation pipeline in Sec. 3.2 to the dataset. We used the Pytorch implementation of AudioUNet [4] and Demucs [5]. We implemented the SEANet generator by ourselves. We used an Adam optimizer and the initial learning rate 0.0001 to optimize each TD-CNN model for 200 epochs with the batch size of 12 and the input duration of 5s.

4.2 Investigation into the Effectiveness of Ground Truth Phase Components

Before delving into the evaluation of TD-PR, we present a preliminary study to show the impact of phase on the artifacts issue of TD-CNN models. In this study, we used SEANet a representative, and replaced the phase of the TD-CNN output with the phase of the corresponding Ground Truth (GT) music, which denoted as TD-CNN w/ GT-phase. Note that GT phase is not available in real world applications.

We then conducted an AB listening test, in which we asked participants to choose the one containing fewer artifacts between the TD-CNN baseline and TD-CNN w/ GT-phase. We selected eleven music pieces for the listening test which cover different periods and styles of different musicians from the MAESTRO test set. Eleven audio pairs are presented in the AB test, in which one pair is for practice and the left ten pairs are for evaluation. Each clip is cut into the duration of 5s. We also regularized the volume of all the samples by Audacity [6]. The input bandwidth for this listening test is set to 3 kHz, as it has been analyzed to be the typical bandwidth of historical recordings [13].

[3] https://github.com/rishikksh20/TFGAN
[4] https://github.com/serkansulun/deep-music-enhancer
[5] https://github.com/facebookresearch/demucs/tree/v2
[6] https://www.audacityteam.org/

4.3 Comparison Between TD-PR and TD-CNN Baselines

TD-PR is proposed to improve the perceptual quality of TD-CNN baselines via phase repair. We evaluated the proposed TD-PR from both objective and subjective aspects. In terms of the objective evaluation, we used the Log-Spectral Distance (LSD) as the metric, which has been widely used in audio SR tasks [1, 2, 4]. LSD is designed as:

$$LSD = \frac{1}{L}\sum_{l=1}^{L}\sqrt{\frac{1}{F}\sum_{f=1}^{F}\left(\log|Y_{l,f}|^2 - \log|\hat{Y}_{l,f}|^2\right)^2}, \quad (5)$$

where $Y_{l,f}$ and $\hat{Y}_{l,f}$ are the ground truth and the estimated magnitude via STFT at l-th time step $(l = 1, ..., L)$ and f-th frequency bin $(k = 1, ..., F)$, respectively.

The subjective evaluation aims at collecting Mean Opinion Score (MOS) from participants to compare the perceptual quality across the input LR music, TD-CNN baseline, TD-CNN w/ TD-PR and ground truth HR music. MOS is commonly used in audio SR tasks to represent the perceptual quality [4, 11]. Participants are asked to rate audio samples according to the similarity with the reference audio, *i.e.*, the groud truth HR music. The range of MOS in our work is set from 1 to 5, where 5 denotes excellent quality (*i.e.*, is the closest to the reference) and 1 denotes bad quality. To avoid auditory fatigue caused by giving too many samples to participants, we evaluated the three TD-CNN models separately in three independent listening tests, which means the MOS values across different tests cannot be directly compared. For each TD-CNN, eleven people with no background in audio engineering participated the listening test. The same eleven music pieces and pre-processing as in the preliminary AB test are used.

5. RESULTS AND DISCUSSION

5.1 Impact of Ground Truth Phase Components

The preference of the AB listening test between TD-CNN baseline and TD-CNN w/ GT-phase described in Sec. 4.2 is shown in Fig. 2. TD-CNN w/ GT-phase is voted to have fewer artifacts with a large margin (95.38% vs 4.62%). Therefore, we concluded that the artifacts in TD-CNN approaches for audio SR tasks is caused by the phase distortion, and the distortion can be repaired by replacing the distorted phase with a more realistic one.

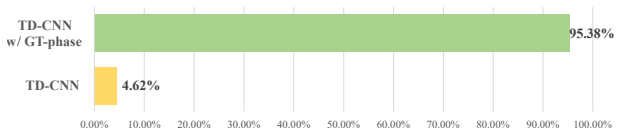


Figure 2. Results of the preliminary AB listening test: 95.38% of the TD-CNN w/ GT-phase is voted to have fewer artifacts.

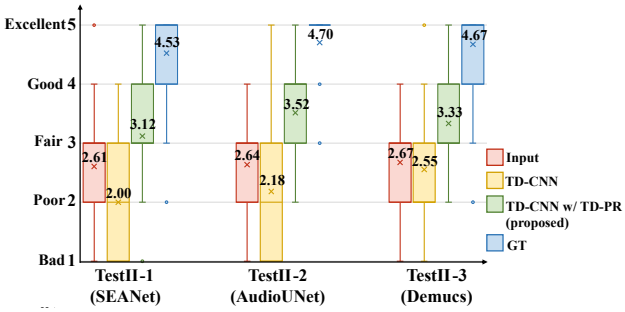


Figure 3. Results of MOS listening test: The box plot of the ratings across input, TD-CNN, TD-PR and GT. TD-PR is applied to three different TD-CNN baselines.

5.2 Results on TD-PR

We conducted the MOS listening test described in Sec. 4.3. The box plot of the MOS test results and the corresponding average for each method are shown in Fig. 3. First, the proposed TD-PR obtained better MOS scores than all three TD-CNN baselines by a large margin, *e.g.*, the proposed TD-PR has higher boxes, and higher average MOS scores of 1.12 (SEANet), 1.34 (AudioUNet), 0.78 (Demucs), revealing that the TD-PR improved the perceptual quality of TD-CNN baselines significantly. Successfully improving three different baselines with a single pretrained vocoder indicates the flexibility of the proposed TD-PR method.

From the perspective of the average MOS scores between input LR music and TD-CNN baselines, it is obversed that TD-CNN baselines obtained lower MOS than the LR input by the deterioration of -0.61 (SEANet), -0.46 (AudioUNet), -0.12 (Demucs). This indicates that the artifacts in TD-CNNs severely harmed the perceptual quality. However, we will show later that TD-CNN baselines obtained better LSD scores (objective metric) than the LR input, indicating that LSD is not a reliable metric to evaluate audio SR and perceptual quality.

In terms of the gap of the average MOS between input and TD-CNN baselines, Demucs shows the smallest gap to the input, which implies that Demucs is the strongest among the three baselines. This observation is also in consistency with its largest parameter amount.

The LSD scores on 4 representative LR bandwidth (2.5 kHz, 3 kHz, 3.5 kHz, 4 kHz) is shown in Table 1. Note

Table 1. LSD results with different input bandwidth and parameter amount of different models.

	2.5kHz	3kHz	3.5kHz	4kHz	AVG	Parameter
Input	2.43	2.19	1.97	1.78	2.09	-
SEANet	0.89	0.78	0.72	0.68	0.77	11M
SEANet w/ TD-PR(proposed)	0.94	0.86	0.82	0.80	0.86	11+6M
AudioUNet	0.83	**0.74**	0.69	0.66	0.73	56M
AudioUNet w/ TD-PR(proposed)	0.89	0.82	0.79	0.77	0.82	56+6M
Demucs	**0.82**	**0.74**	**0.68**	**0.64**	**0.72**	134M
Demucs w/ TD-PR(proposed)	0.89	0.83	0.79	0.77	0.82	134+6M
Ground truth	0	0	0	0	0	-

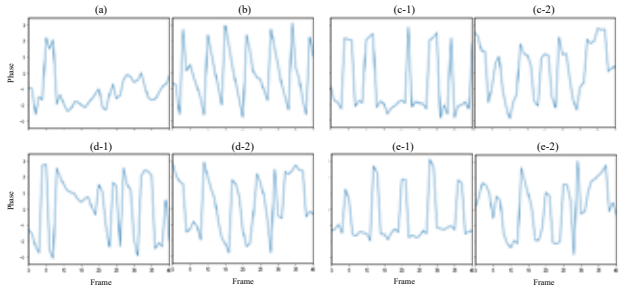


Figure 4. Visualization of a set of phase spectrograms: (a) low-resolution input; (b) ground truth; (c-1) SEANet; (c-2) SEANet w/ TD-PR (proposed); (d-1) AudioUNet; (d-2) AudioUNet w/ TD-PR (proposed); (e-2) Demcus; (e-2) Demucs w/ TD-PR (proposed).

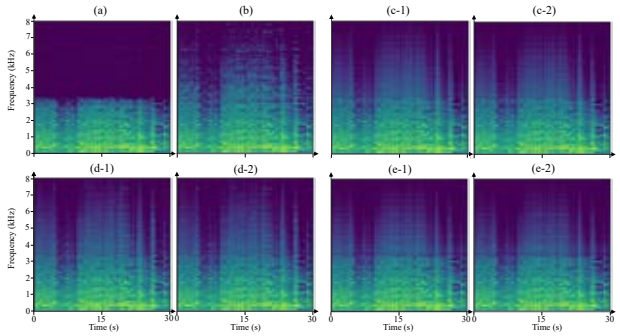


Figure 5. Visualization of a set of magnitude spectrograms: (a) low-resolution input; (b) ground truth; (c-1) SEANet; (c-2) SEANet w/ TD-PR (proposed); (d-1) AudioUNet: (d-2) AudioUNet w/ TD-PR (proposed); (e-2) Demcus; (e-2) Demucs w/ TD-PR (proposed).

that the proposed method can deal with any bandwidth between 2.5 kHz and 4 kHz. The results show that both TD-PR and their TD-CNN baselines got much lower LSD than LR input, indicating that music SR is successfully achieved. Although the proposed method got sightly worse LSD scores than the baselines, we argue this is trivial, because the aforementioned MOS listening test revealed a significant gap in perceptual quality between TD-PR and baselines. Although LSD can well reflect how well the high frequency magnitude is recovered in each model, it can't reflect the degree of the phase distortion and has been observed not highly correlated with perceptual audio quality in previous literature [4].

5.3 Qualitative Evaluation of TD-PR

We visualize a part of phase spectrograms in Fig. 4 and their corresponding magnitude spectrograms in Fig. 5 to qualitatively evaluate the proposed TD-PR method. The visualizations include the spectrograms of LR input, ground truth, three TD-CNN baselines and their corresponding TD-PR outputs. For a clear view in Fig. 4, we plot only the phase of a single frequency bin for the first 40 time frames of an audio sample, as the phase spectrogram across multiple frequency bins is difficult to understand. The visualizations reveal that the proposed TD-PR successfully produced a phase distribution that is closer to ground truth's compared to TD-CNN baselines. Meanwhile, as TD-PR only repairs the phase components, we cannot observe significant differences in magnitude spectrograms shown in Fig. 5. Nevertheless, perceptual quality is improved significantly by TD-PR. The visualizations again validate that phase distortion has been the cause of the annoying artifacts in TD-CNNs.

6. CONCLUSION

In this research of music Super-Resolution (SR), we delved into Time-Domain Convolutional Neural Networks (TD-CNNs), trying to identify the cause of the annoying artifacts and improve TD-CNNs' perceptual quality by alleviating the artifacts. To the best of our knowledge, this work is the first to demonstrate the artifacts in TD-CNNs are caused by the phase distortion via a subjective experiment. We further propose Time-Domain Phase Repair (TD-PR), which uses a neural vocoder pretrained on the wide-band data to repair the phase components in the waveform output of TD-CNNs. The proposed TD-PR achieved better mean opinion score, significantly improving the perceptual quality of TD-CNN baselines. Moreover, a single pretrained vocoder can be directly applied to arbitrary TD-CNNs without additional adaptation. Since the proposed TD-PR only repairs the phase components of waveform, the improved perceptual quality in turn indicates that phase distortion has been the cause of the annoying artifacts of TD-CNNs. The findings and comprehensive evaluations presented in this work offer a new perspective for the future improvement of audio super-resolution algorithms. This work inspires us to combine the advantages of TD-CNNs and neural vocoders in future, to develop a model that can better address the challenges in music super-resolution.

7. REFERENCES

[1] V. Kuleshov, S. Enam, and S. Ermon, "Audio super-resolution using neural networks," in *ICLR (Workshop Track)*, 2017.

[2] S. Hu, B. Zhang, B. Liang, E. Zhao, and S. Lui, "Phase-Aware Music Super-Resolution Using Generative Adversarial Networks," in *Proc. Interspeech 2020*, 2020, pp. 4074–4078.

[3] Y. Li, M. Tagliasacchi, O. Rybakov, V. Ungureanu, and D. Roblek, "Real-time speech frequency bandwidth extension," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 691–695.

[4] H. Liu, W. Choi, X. Liu, Q. Kong, Q. Tian, and D. Wang, "Neural vocoder is all you need for speech super-resolution," *arXiv preprint arXiv:2203.14941*, 2022.

[5] T. Y. Lim, R. A. Yeh, Y. Xu, M. N. Do, and M. Hasegawa-Johnson, "Time-frequency networks for

audio super-resolution," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 646–650.

[6] H. Wang and D. Wang, "Towards robust speech super-resolution," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 29, pp. 2058–2066, 2021.

[7] S. Sulun and M. Davies, "On filter generalization for music bandwidth extension using deep neural networks," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 1, pp. 132–142, 2021.

[8] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 4395–4399.

[9] M. Tagliasacchi, Y. Li, K. Misiunas, and D. Roblek, "Seanet: A multi-modal speech enhancement network," *arXiv preprint arXiv:2009.02095*, 2020.

[10] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Demucs: Deep extractor for music sources with extra unlabeled data remixed," *arXiv preprint arXiv:1909.01174*, 2019.

[11] J. Su, Y. Wang, A. Finkelstein, and Z. Jin, "Bandwidth extension is all you need," in *ICASSP 2021*. IEEE, 2021, pp. 696–700.

[12] Q. Tian, Y. Chen, Z. Zhang, H. Lu, L. Chen, L. Xie, and S. Liu, "Tfgan: Time and frequency domain based generative adversarial network for high-fidelity speech synthesis," *arXiv preprint arXiv:2011.12206*, 2020.

[13] E. Moliner and V. Välimäki, "Behm-gan: Bandwidth extension of historical music using generative adversarial networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 943–956, 2022.

[14] R. Yamamoto, E. Song, and J.-M. Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP 2020*. IEEE, 2020, pp. 6199–6203.

[15] C. J. Steinmetz and J. D. Reiss, "auraloss: Audio focused loss functions in PyTorch," in *Digital Music Research Network One-day Workshop (DMRN+15)*, 2020.

[16] C. Hawthorne, A. Stasyuk, A. Roberts, I. Simon, C. Huang, S. Dieleman *et al.*, "Enabling factorized piano music modeling and generation with the maestro dataset," *arXiv preprint arXiv:1810.12247*, 2018.

FLOW: A MUTITIMBRAL, POLYPHONIC, MODULAR, ADDITIVE MUSIC SYNTHESIZER

Sean LUKE[1]

[1] *Department of Computer Science*, **George Mason University**, Washington, DC USA

ABSTRACT

Flow is an open source additive synthesizer that is very capable, but is specifically designed to be programmed by a musician without requiring the extraordinary number of parameters commonly demanded by additive synthesis. To achieve this, Flow works largely like a modular synthesizer, except that modules pass arrays of partials to one another rather than audio. Flow is polyphonic and multitimbral, supports MPE and microtonality, and easily extended and hacked. This paper describes Flow, its motivation, and its architecture, and discusses some of the lessons learned in its development and use as an instrument.

1. INTRODUCTION

Flow is a large additive software synthesizer with an unusual combination of features. It is polyphonic and multitimbral. It has full support for MIDI features including MPE and microtonality. It allows for up to 256 sine waves (partials) with both real-valued frequency and amplitude, but not phase. But most importantly, it is structured as a fully modular synthesizer, including cyclic patch connections, and patches can have any number of modules. Flow has over 70 modules available. Patches may contain special modules which encapsulate entire other patches (so-called *macros*). Flow is written in Java and is free open source, available at https://github.com/eclab/flow

I began Flow in 2018 with a specific experimental hypothesis in mind: is it possible to make a sophisticated additive synthesizer which does not overwhelm the musician with a massive number of required parameters? I think the answer has proven to be yes.

An additive synthesizer commonly changes a large number N of sine waves dynamically in real time, and each of these waves may have some M parameters defining how it changes over time, and so the total number of parameters can get large very quickly. Thus additive synthesis is notorious for its high number of parameters and tedious programming.

Rather than control each wave independently, Flow instead treats the entire set of waves as a group and passes them through a graph structure of modules. Each module sculpts the waves in different ways, ultimately outputting the result. You might recognize this architecture: it is the architecture commonly used by subtractive modular synthesizers. However, Flow's modules are rapidly passing arrays of partials to one another rather than sound waves, and are operating on the sound solely in the frequency domain, not the time domain. In essence, Flow is applying a classic modular subtractive interface to additive synthesis.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. HISTORY AND RELATED WORK

Additive synthesis is very old. Indeed, organ ranks may be considered mechanical additive synthesizers, with each rank contributing new partials to the final sound.

Early additive synthesizers were electromechanical. Among the very most important early synthesizers was Thaddeus Cahill's extraordinary Telharmonium, a massive additive synthesizer circa 1896. The Telharmonium used rotating tonewheels and magnetic dynamos (pickups) to sum harmonics and broadcast the resulting sound to remote listeners. Figure 1 illustrates the tonewheels. Many of these concepts have since found their way into the Hammond Organ, which even now applies tonewheels and pickups to produce its sound.

In 1964 James Beauchamp developed the Harmonic Tone Generator [1] at the University of Illinois, Urbana-Champaign. This instrument used analog electronic circuits to synthesize a tone of six harmonics. Much electronic additive synthesis since then has been digital, and has often been associated with the analysis of sounds in addition to their synthesis. Starting in 1967, Jean-Claude Risset used Bell Labs's Music V software [2] to analyze and reproduce trumpet and later bell sounds [3]. The phase vocoder [4], also developed at Bell Labs, has likewise served as an analysis and additive resynthesis tool [5].

Notable early examples of digital additive synthesis for music production included the RMI Harmonic Synthesizer, the Fairlight Qasar M98 and CMI, and the New England Digital Synclavier II. In the 1980s and 1990s, Teisco/Kawai was perhaps the biggest producer of additive synthesizers, including the K3, K5, and K5000 series of synthesizers, with increasingly powerful additive synthesis capabilities. For further discussion of these and other examples, see [6].

Nowadays additive synthesis is less common and largely relegated to software. There are still many examples however, including Native Instruments's Razor, Image-Line's Harmor, Camel Audio / Apple's Alchemy, and AIR Music Technology's Loom. Additive synthesis is also popular in the open source community, perhaps because simple versions of the technique are easy to implement.

Many software synthesizers take a classic approach to additive synthesis, attaching modulators to each of the individual harmonics or partials. But a few, including some of the recent software synthesizers mentioned earlier, take an approximately modular approach which to some degree resembles Flow. Probably the most similar to Flow is AIR's Loom [7], in which a fixed number of processing modules are arranged in series. Each module receives an array of partials, modifies them in some way, and passes them down the line to the next module. Thus unlike Flow (as discussed later) there is only a total ordering among the modules. Each of these modules may then be modulated by a fixed set of separate modulation functions.

3. ADDITIVE SYNTHESIS IS HARD TO USE

Additive synthesizers usually produce sound by continuously changing the features of some N sine waves, or *partials*, and outputting their sum. These sine waves may differ in frequency, amplitude, and phase, and all three of these characteristics can change dynamically with time. In theory, if N is sufficiently large, and the change over time is sufficiently rapid, an additive synthesizer can produce any sound with adequate fidelity.

From an algorithmic standpoint, additive synthesis is trivial: we're just adding sine waves. But from the *musician's perspective,* additive synthesis can be very hard, as it can present a huge parameter space. This is because in its general form, additive synthesis requires at least $3N$ time-varying functions (frequency, amplitude, and phase for each of its N partials), each with sufficient parameters to describe the change over time of a partial when played.

Some simple additive synthesizers require only a few partials, but to make rich or complex sounds it is not uncommon for N to be large, often in the hundreds, particularly to adequately describe low tones. This can result in a very large number of parameters to program. For example, a single patch for the Kawai K5000 series — which were well regarded additive synthesizers — could have up to 6000 parameters.

There is also the computational cost to consider. These N sine wave functions must be calculated in real time per voice, which can be expensive, though in certain cases depending on the nature of the architecture, and the limits on the sound generated, this cost can be reduced by using a Short-Time Fourier Transform.

I think that these two challenges are the main reasons why additive synthesis has not been as popular as other methods. And these challenges are so dominant that many additive synthesizer designs are forced to reduce the parameter space in one way or another. Additive synthesizers generally use one or more of three strategies to do this.

First, it is very common for additive synthesizers to constrain the frequency of each sine wave to be an integer multiple of the fundamental. That is, they work not on partials but on arrays of *harmonics.* This simplification is justified under the assumption that sounds perceived as tonal are often largely composed of harmonics: noise or anharmonic sounds might be added in after the fact in a traditional subtractive manner.

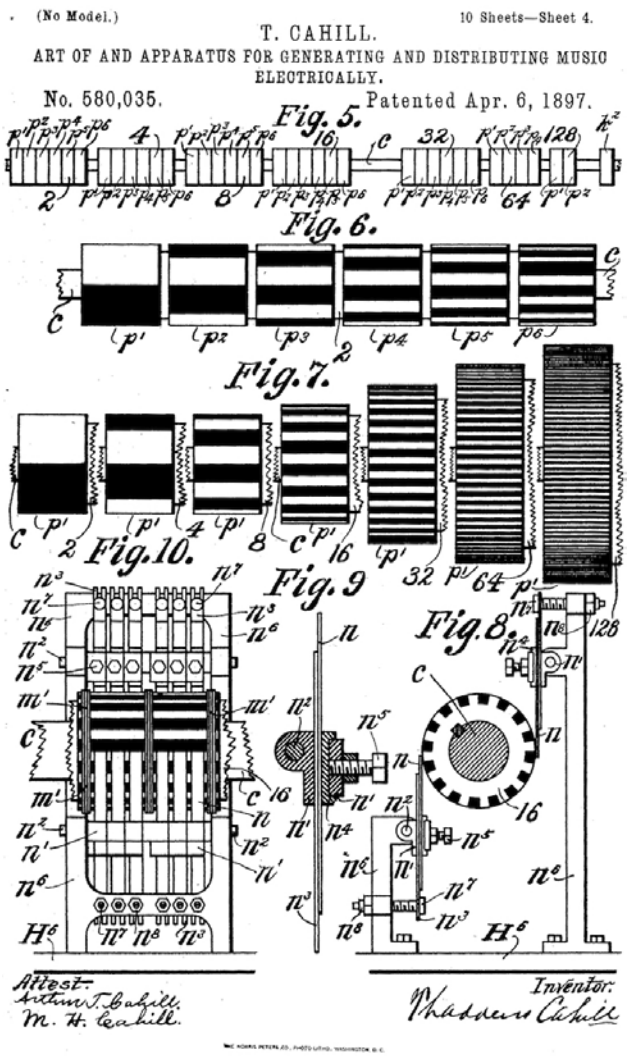


Figure 1. Illustration from 1897 Telharmonium patent filing, showing additive tonewheels.

Second, many additive synthesizers discard the phase of each sine wave in a voice, resetting it to 0 each time a new note is played. This simplification is justified under the presumption that, at least for the higher frequency sounds, it can be difficult to distinguish between two sounds which only differ in the (arbitrary) phases of their respective partials. Furthermore, if synthesizers are constrained to just produce harmonics, they cannot get out of sync with one another in phase, and so starting at 0 is reasonable.

Third, synthesizers might reduce the total number of partials, or skimp on the complexity of the modulation used to manipulate each of them. For example, let us restrict ourselves to just the Kawai synthesizers: while the Kawai K5 had 126 harmonics per voice, the K5000 series only had 64, though it could overlay multiple sets of harmonics in some situations. The K3 had just 32 harmonics, which is not atypical for many simple additive synthesizers. On the other hand, while the K5000's harmonics each had their own envelopes, the K5's harmonics had to share just four of them.

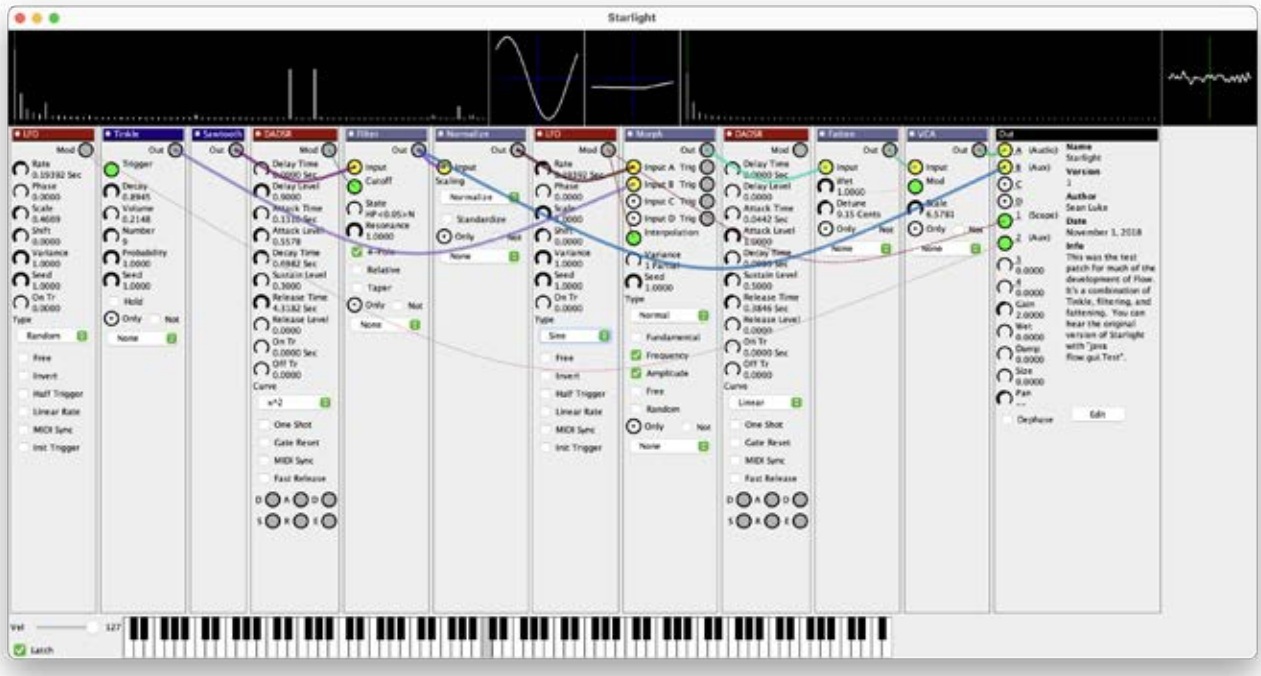


Figure 2. Flow interface, showing the partials displays, oscilloscopes, and sound output at top, the modules of a patch at center, and the optional keyboard at bottom.

4. FLOW

Flow takes a different approach. Rather than modulate the parameters of individual partials over time (though it can do that), Flow passes whole sets of partials through modular functions which manipulate them as a group. Flow is very much like a classic modular synthesizer, except that the modules work in the frequency rather than time domain: they do not pass sound to one another, but rather arrays of partials. Modules can be connected in any way the musician sees fit. Flow presently supports:

- Up to 256 partials, user-selectable.
- Partials with floating-point frequency and amplitude. Phase is discarded.
- Any number of modules per patch.
- 1–32 voice polyphony, user-selectable.
- Full multitimbrality. Only one timbre at a time may be manipulated in real-time in the UI.
- Sampling rates up to 48KHz, at 16 bit depth.
- MIDI Polyphonic Expression (MPE) [8], Clock Sync, Channel and Polyphonic Aftertouch, Pitch Bend, NRPN, RPN, and CC.
- Microtonal architectures via Scala files [9].

The idea is that, rather than manipulate each partial, the musician can select a small set of functions to manipulate the partials in a predictable way, which presents a much simpler and more intuitive interface. However, this combination is also computationally expensive. However it also presents a much easier environment for the musician to construct additive patches.

5. INTERFACE

Flow presents itself as a modular software synthesizer in a fashion similar to VCV Rack and similar [10]. An example of Flow's interface is shown in Figure 2. At the top of the window are two (long) displays for showing the frequency and amplitude of partials in a sound: the one at left displays the partials of the sound being generated. Also, (centered) are two oscilloscopes for modulation signals, and (far right) one oscilloscope for the final sound output. These displays are optional. Also optional is the small keyboard at bottom.

A Flow patch consists of some *N modules*, each connected via virtual patch cables. In the center of the window (Figure 2) are the patch's modules, arranged left-to-right first-to-last. Modules may be added, removed, and changed in order relative to one another.

5.1 Patch Cables and Patching

There are two types of patch cables, and they connect only to input and output sockets of their type. First there are *sound cables* (drawn with thick lines) along which sound data is transferred in the form of arrays of 256 partials. Each partial consists of a frequency, an amplitude, and an *ordering*, a unique number 0...255 which represents the sine wave generator in the Output's generator bank associated with that partial (see Section 8.4). Partials in the array

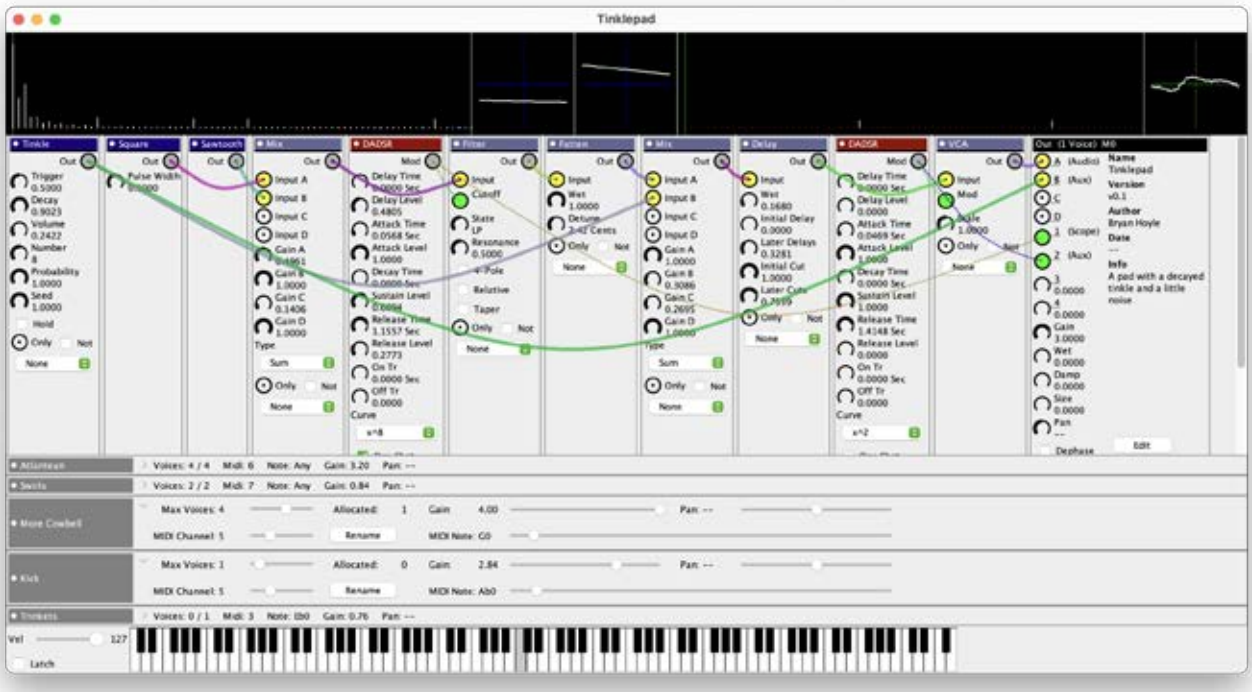


Figure 3. Flow interface, showing the modules of the primary patch and five multitimbral subpatches below.

are always sorted by frequency, as the large majority of modules' algorithms benefit from frequency ordering. Flow pulses each of its modules in order left to right, and when a module is pulsed, it retrieves the latest incoming partials from modules connected to it, then produces a new array of partials for other modules to request from it. Flow attempts to pulse all the modules as quickly as possible.

Second, there are *modulation cables* (drawn with thin dashed lines), along which modulation — essentially control voltage — data is passed from module to module. Flow offers two twists on the traditional modulation cable. First, these cables are not plugged into jacks per se, but directly into the modulation dials on each module. For example, a Filter module might have a cutoff frequency dial, either settable directly by the musician, or pluggable with a cable to control it instead. If it is useful for a parameter to be both modulated and user-controlled, the module simply provides two dials. Second, modulation cables carry not only modulation signals but also one or more typed *triggers* which signal events which have occurred. Triggers are used for many purposes. For example, an LFO module might output along its cable not only the LFO signal itself, but also a trigger fired each time the LFO has completed one period: this could then be attached to a sequencer to clock it. Modules request data along modulation cables in the same manner as is done for partials.

Flow rarely uses gates. This is because all modules directly receive cooked MIDI event information, including notes, control change and pitch bend information, etc. The most common use of gates in a modular synthesizer is to provide note on/off, which Flow's modules already know about.

Normally cables are forward-patched, meaning that there is a partial order of data flow among the modules from early (leftmost) modules to later (rightmost) ones. However nothing precludes a musician from back-patching: connecting a cable to flow data from a module to an earlier one to its left. A common use of back-patching is to provide delayed feedback loops. Thus Flow's patching structure can accommodate any cyclic directed graph. Back-patching is handled in the obvious way, namely that when the upstream earlier module requires partials or modulation data from the downstream module, it receives the previous data from the *last* iteration.

Flow has one special module called *Out*, of which exactly one copy always exists in a patch. Out is a sink which receives partials data and hands it off to the system to process into sound.

Though the musician only perceives one set of modules, in fact Flow maintains an independent set of modules for each voice to provide for polyphony.

6. MULTITIMBRALITY

Flow is not only polyphonic but also multitimbral. In a classic multitimbral synthesizer there exist both *single-mode* and *multimode* patches. A single-mode patch describes a single kind of sound. A multimode patch contains multiple different single-mode patches, or references to them, and states how voices and other performance elements are to be allocated to the various single-mode patches.

Flow's approach is a little different. Flow patch files consist of a *primary* (single-mode) patch, and zero or more (single-mode) *subsidiary patches*. Both the primary and

subsidiary patches have voices, MIDI data, and the like allocated to them, but only the primary patch can be edited and manipulated by the musician during performance. However the musician can, at any time, swap the primary patch with any subsidiary patch, so the subsidiary patch can now be manipulated instead. Subsidiary patches can be loaded or discarded from the main patch at any time.

7. MACROS

One of Flow's most unusual features is its *Macro* facility. After you have created a patch and saved it to disk, it is possible to load that patch and encapsulate it into a single module (a Macro), to be used among other modules in another patch.

The Out module contains not just a single jack to receive output partials, but three more auxiliary jacks of this kind, as well as four modulation jacks. All eight of these jacks can be named, and so when a patch is loaded as a Macro, the Out jacks connected in the patch are exposed as the Macro's outputs. Furthermore, the patch may have one or more copies of an optional module called *In* which has eight nameable input partials jacks and eight nameable input modulation jacks, plus default modulation values for them. If connected, these then appear as inputs to the Macro.

Macros are pulsed in a depth-first fashion: when the Macro module is pulsed, it loads the needed data for its *In* module, then recursively pulses all of its internal modules left-to-right once, including recursive nested Macros, then prepares resulting data from its *Out* module to provide to downstream outer modules.

Flow comes with a number of examples of Macros as useful utilities. These include formant voices, sound fatteners, brick wall filters, random drift utilities, bad attempts at supersaws, etc.

8. MODULES

Excepting Macros, Flow at present has 71 modules divided into five categories. The title bar color of each module reflects its category (see Figure 2). It is straightforward to add new modules to Flow, and the manual has step-by-step tutorials explaining how to do it.

I will not here subject the reader to the tedium of enumerating all these modules, but will highlight a few as examples.

8.1 Partials Generators

These are the rough equivalent of oscillators in a typical synthesizer. Flow has 20 partials generators, producing arrays of partials from common waves (like sawtooth), or from modulatable wavetables and vocoders, from musician-specified partials or harmonics, from time-varying functions that introduce and remove partials, and so on. Flow also sports modules producing partials derived from the waves or harmonics of other historic synthesizers, namely the Casio CZ series, Kawai K3 and K5, Ensoniq SQ-80, and Sequential Prophet VS, as well as those from the AdventureKid Waveform collection. [11]. Note that though the original data for some of these modules is derived from time-domain waves, these modules are all purely additive and frequency-domain.

8.2 Partials Modifiers

Flow has 28 modules which take partials as input, modify them, and output the result. These include a variety of time-varying filters, amplifiers, and mixers or combiners, but also many other modifiers special to Flow's additive nature.

Filters are particularly interesting. Among its other filters, Flow has modules which model N-pole filters, but does not have to restrict itself to Z-domain style digital filters. It can model a Laplace-domain (analog) filter clear to the Nyquist frequency, simply by applying the filter's amplitude response function to each partial. The phase effects of filters are, however, ignored.

Flow also has several modules dedicated to taking multiple additive partials streams and combining them in some modulatable fashion. This includes everything from cross-fading to morphing the frequencies of partials to partial-by-partial dissolves.

Combination is nontrivial because Flow has a fixed number of total partials, and so when two partials streams are combined, we must decide which partials don't make the cut. The combiners vary largely in the strategy used to select partials, or to gradually introduce partials and remove others under modulation. Additionally, when a partial is removed and another is introduced, the corresponding sine wave generator is lurched from the frequency and amplitude the first partial to the second, and in so doing combiners must employ schemes to prevent these changes from producing pops and other unwanted artifacts.

Flow also contains modules peculiar to the frequency domain, such as skeletonization and dilation, fattening (duplicating and detuning specific partials), adding frequency or amplitude jitter to them, shifting and rotating select partials, as well as delays and transition smoothers, and so on.

Nearly every partials modifier module can be *constrained*, that is, restricted as to *which* partials it is permitted to operate on. In many cases this is as simple as performing a function on every partial, then restoring certain partials to their previous values. In many other situations, notably combination modules, constraints are more complex.

8.3 Modulation Generators and Modifiers

Flow has 17 modules dedicated to generating, or modifying, modulation signals. These include the usual suspects such as LFOs, several envelope generators, sequencers, joysticks, sample and hold, and the like. They also include modules that expose a variety of MIDI features and event data not used by modules by default, including MPE modulation, NRPN, CCs, etc.

8.4 Utility Modules

The remaining modules are prosaic: *Out* and *In* as described before, plus modules for stretching patch points, adding textual notes, locking sounds to specific pitches, and so on.

9. THREAD ARCHITECTURE AND SOUND GENERATION

Flow consists largely of two parts: voice management and additive output. For each and every voice, the voice manager maintains a separate copy of modules in order to produce the latest partials of the sound. These partials are then handed to the additive output, which maintains its own independent bank of threads to push the partials though sine wave functions, add them up, and add effects.

Given that Flow is multitimbral and polyphonic, it should not then be surprising that Flow is very heavily multithreaded. As shown in Figure 4, Flow consists of a *MIDI Input Receiver* thread, which receives and processes all incoming MIDI data. This is passed to the *Voice Sync* thread, which manages per-voice processing of each of the modules. Each voice has its own set of modules and works with a (possibly shared) thread to usher partials from one module the next, and ultimately to the *Out* module. The Out module serves as the interface between the Voice Sync thread and the *Sound Output* thread. Once the latest partials from all voices have all arrived, the Sound Output thread takes the partials and updates the sine waves being used to produce the sound. This update is done with a leaky integrator so as to not produce artifacts stemming from a sudden change in frequency or amplitude. The Sound Output thread maintains a bank of output threads which divvy up the current sine waves and continually produce the latest sine output values. The sum of these values is then pushed through a Freeverb reverb implementation and output as sound.

The GUI also has its own separate event thread: this thread largely interacts with the Voice Sync thread to change module parameters and architecture; though at times it does work with the MIDI Input Receiver and Sound Output threads.

The Voice Sync threads are asynchronous from the Sound output threads: even if the patch's modules take a long time to run, and so the partials are slow to arrive, the output threads still work in the background to keep the sound buffer filled. Flow can be asked to try to maintain a rough ratio of output samples to voice sync partial updates. By default this is normally around 16 to 32 samples generated per partials update.

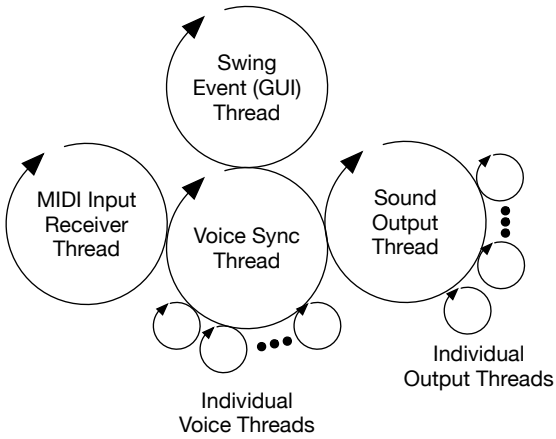


Figure 4. Flow's primary threads. Touching circles implies locking interaction.

10. DISCUSSION

I think Flow has been very successful as a demonstration of how to reduce the parameter complexity of additive synthesis. I and others have long used it as a performance and sound design tool, but it has also been particularly useful to me in an educational context, illustrating how sound synthesis works in the frequency domain.

Development and subsequent use of Flow has nonetheless brought up interesting issues worth discussing here.

10.1 Consequences of Disregarding Phase

Like many additive synthesizers, Flow discards phase as a computational and interface simplification measure. However while additive synthesizers constrained to harmonics do not need to concern themselves with changes in relative phase, Flow's partials can change in frequency in real-time. Consider two sine waves of the same frequency and phase, playing in unison. Flow modifies the frequency of one of them by some arbitrary amount, waits for a moment, then returns the frequency to its previous value. Now the two waves are out of phase.

For higher frequency notes, such as above middle C, the difference in sound can be negligible. But this is not the case for lower notes. The ability of humans to discern the difference between in-phase and randomized-phase partials grows dramatically with lower frequency [12]. Particularly in the case of very low frequencies, highly synchronized sounds such as square waves sound very different when their respective partials are out of phase with one another, not the least of which is because of the sudden and large jump in the sound wave (such as the vertical portion of a sawtooth) when all the phases are in sync.

Flow cannot compensate for patches which shift the phases of their underlying partials: the change in sound will be part of the nature of the patch's timbre. However it does reset their phases when a new note is played. Furthermore, Flow patches have an option ("Dephase") to add to each partial a specific deterministic random phase amount, and this masks most perceived phase changes when individual partials' frequencies are modified. In my experience, dephasing a low-frequency sawtooth or square wave also makes it sound "smoother", less harsh, and somehow more electric. This may or may not be desirable.

Though Flow gets along fine without phase, sometimes phase would be helpful, particularly in sampling sounds and reproducing them, such as with a phase vocoder. Flow's *Wavetable* module can record sounds (speech, say), chop them up into individual harmonics, and reproduce them, but as it lacks phase information it can only do so in a robotic fashion reminiscent of old-style vocoders.

10.2 Technical Issues

Flow's architecture brings up a number of technical issues. Here are three.

10.2.1 Number of Partials

Flow at present has only 256 partials, which is fewer than commercial additive synthesizers (for example, 384 in Harmor or 512 in Loom). This is not due to a technical limitation. The only reason for this is that Flow uses a single byte as the ordering ID for a partial, that is, the tag that identifies which sine wave generator is assigned to the partial. In the future Flow may move to a short or int, which would allow far more partials.

10.2.2 Partial Update Rate

Flow's per-voice sync threads are typically set to be slower than its output threads: as mentioned before, it's common for about 16 to 32 samples to be generated per partials update, though this can be changed if you have a fast enough computer. Furthermore, after new partials arrive the current partials are gradually adjusted to match them using a leaky integrator. This introduces latency.

This disconnect also has curious effects on extremely short sounds, particularly bursts of noise such as for a click or snap. Because the onset of the click is not synced to *when* the next set of partials is updated, clicks will not be consistent in amplitude. Additionally, because partials are not updated 1:1 with samples, modeling FM synthesis etc. would be difficult.

10.2.3 Use of Java

Flow is written in Java, and this allows Flow to be easily ported to a variety of platforms. While Java is a very fast language, one of its slowest areas is array access, and Flow relies heavily on array access for its partials. Nonetheless I have found Java more than sufficient to handle Flow's computational requirements.

Java is a garbage collected language and does not provide much control over how or when it happens. Java's traditional garbage collectors may occasionally produce a glitch in the sound output as Flow is unable to fill the sound buffer while being garbage collected. However Java's new low-latency collectors, such as its Z collector, entirely eliminate this issue.

The primary disadvantage of Java is that it is generally incompatible with the VST API; as a result Flow will likely remain a standalone application rather than embeddable in a DAW.

10.3 Specific Value of Additive Synthesis

Flow has a many useful modules which are peculiar to additive synthesis. For example, *Dissolve* will cross-fade from one sound to another but will do so by replacing random partials one-by-one in one sound with the other. *Jitter* will add random noise to the amplitude and/or frequency of select partials. *Shift* will, among other things, change the amplitude of each partial to that of the partial immediately below it in frequency. And so on.

However it is also true that Flow contains other modules which are essentially additive versions of subtractive classics, such as filters, amplifiers, mixers, and oscillators producing traditional waves. And I have found that, in my own patch development, I do often use these modules. The open question is why. Is it because I am used to these sounds historically and so sometimes gravitate to them? Is it because they are easy to understand and integrate? Or is it because many designed-for-additive modules can result in significant anharmonicity or manipulation of high frequencies?

Flow was developed partly in the hope that additive synthesis would offer sounds or programming approaches which were fundamentally different from subtractive synthesis designs. I strongly believe it does. But the question as to what degree this is true is worth examining further.

11. CONCLUSION AND FUTURE WORK

Flow was originally conceived as part of an effort to develop an additive synthesizer with sophisticated patching, modulation, and partial-manipulation capabilities, but which did not impose on the musician the tedium of programming a large number of parameters. To this end, Flow has adopted a modular synthesis approach, transferring entire sets of partials from module to module to be processed. While this pipeline approach has been adopted occasionally by other software, Flow goes rather further along this route, providing arbitrary connection graph structures including ones with cycles, any number of modules, and recursive module development in the form of macros.

Even though Flow does not require large numbers of parameters, it still has a very large array of options, because it has many modules and the musician can arrange and connect any number of them in any order. This makes attractive the notion of automating the patch-development process in order to search the space of patches more effectively.

I have previous work in automated patch exploration and co-creative tools. I have developed a large patch editor tool for many synthesizers, called Edisyn [13]. Part of Edisyn is a facility which uses evolutionary computation methods (such as the Genetic Algorithm) to work with the musician to explore the space of patches without having to program them. In this approach, known as *interactive evolution*, the program auditions some N patches to the musician, who selects those he prefers. Based on this information, the program mixes and matches, then slightly mutates, these patches to create N new patches in the vicinity of the musician's selections, then auditions them to the musician, and so on. Essentially, the program is co-creatively working with the musician to wander through the space of patches towards those he likes best.

This technique exploits the fact that the large majority of synthesizer patches may described as fixed-length arrays of numbers. But Flow's patches are arbitrary graph structures, which presents a large and non-metric space without a proper distance measure. Such spaces are difficult for evolutionary computation and other stochastic optimization methods to tackle efficiently, though there are certain approaches, such as genetic programming, which might be applied [14]. As future work I am interested in asking how these methods might be used, and to what degree, to simplify the task of programming additive synthesizers even more.

Acknowledgments

Microtonality and MPE were added to Flow with help from Vi Hoyle. Thanks to Giorgio Presti and Federico Avanzani at LIM (U. Milano) for their support and feedback in the development of this paper.

12. REFERENCES

[1] J. W. Beauchamp, "Additive synthesis of harmonic musical tones," *Journal of the Audio Engineering Society*, vol. 14, no. 4, pp. 332–342, October 1966.

[2] M. V. Matthews, *The Technology of Computer Music*. MIT Press, 1969.

[3] J.-C. Resset, "Computer music experiments, 1964–..." *Computer Music Journal*, vol. 9, no. 1, pp. 11–18, Spring 1985.

[4] J. L. Flanagan and R. M. Golden, "Phase vocoder," *Bell System Technical Journal*, vol. 45, no. 9, pp. 1493–1509, 1966.

[5] J. A. Moorer, "Signal processing aspects of computer music — a survey," *Computer Music Journal*, vol. 1, no. 1, pp. 4–37, February 1977.

[6] S. Luke, *Computational Music Synthesis*, 1st ed., 2021, available for free at http://cs.gmu.edu/~sean/book/synthesis/.

[7] Air Music Technology, "Loom II," Available as https://www.airmusictech.com/virtual-instruments/loom-ii.html, as of 1/1/2024.

[8] *MIDI Polyphonic Expression (Version 1.0)*, MIDI Manufacturers Association, 2018.

[9] M. O. de Coul, "Scala," Available at https://www.huygens-fokker.org/scala/, as of 1/1/2024.

[10] "VCV Rack Modular SoftSynth," Available at https://vcvrack.com/, as of 1/1/2024.

[11] K. Ekstrand, "Adventure Kid Waveforms (AKWF)," Available at https://www.adventurekid.se, as of 1/1/2024.

[12] M.-V. Laitinen, S. Disch, and V. Pulkki, "Sensitivity of human hearing to changes in phase spectrum," *Journal of the Audio Engineering Society*, November 2013.

[13] S. Luke, "Stochastic synthesizer patch exploration in Edisyn," in *International Conference on Computational Intelligence in Music, Sound, Art and Design (EvoMUSART)*, 2019.

[14] ——, *Essentials of Metaheuristics*, 2nd ed. Lulu, 2013, available for free at http://cs.gmu.edu/∼sean/book/metaheuristics/.

LOOM: A SPECULATION OF ORAL TRADITION IN IMMERSIVE MEDIA

Yingjia GUO (lemonguo@stanford.com) [1] and **Mengtai ZHANG** (mengtaizhang@gmail.com) [2]

[1]**Stanford University**, Stanford, CA, US
[2]**Fluffy Cactus**, New York, NY, US

ABSTRACT

Through an analysis of our in-progress Virtual Reality opera "Loom", this paper explores the potential and limitation of immersive media in its intersection with musical heritages which are passed down as oral traditions. "Loom" is based on our fieldwork in the villages of Kam people, an ethnic minority group in the southwest of China, whose polyphonic group singing tradition has been passed down orally for hundreds of years. Given the shared prominence of immersion and embodied experience in the practice of Kam music and in the medium of Virtual Reality, this paper proposes the creation of immersive media as a way to engage with and speculate on oral traditions.

1. INTRODUCTION

In recent years, immersive media, an emerging field of audiovisual technologies and modes of perception known as Virtual Reality (VR), Augmented Reality (AR), Mixed Reality (MR), and various related configurations, have attracted artists and researchers who engage in the documentation and speculation of oral traditions [1–3]. Approaching older traditions with new technologies, the interaction of such seemingly contrasting time continua reveal shared challenges and potentials in both paradigms of experience and transmission. Broadly speaking, both mediums foreground a sense of presence and immersion through embodied experiences, while negotiating displacement resulted from their underlying technical apparatuses and sociocultural conditions. In this paper, we will discuss our Virtual Reality opera "Loom", a work based on our field research of the musical tradition of the Kam people in southwest China, to reflect on the intersection of these two mediums, their possibilities, and limitations.

2. PRESENCE AND DISPLACEMENT IN THE PRACTICE OF CONTEMPORARY KAM GRAND CHOIR

The Kam, or known as Dong (侗族) in Chinese, constitute one of the 55 ethnic minorities officially recognized in China. They mainly inhabit Guizhou, Hunan and Guangxi Provinces in southwestern China [4]. Still the dominant language used in the Kam areas, the Kam language has no widely used written form [4], and the singing of Kam songs has traditionally served the important role of transmitting knowledge across generations [5].

Da ge (大歌), known in English as "Big Song" or "Grand Choir", or ga lao in the Kam language, is a category of Kam vocal music, characterized by polyphonic singing performed by single-sex groups, without instrumental accompaniment. The most well-known form of Kam music outside of the Kam community, the Kam Grand Choir was recognized as an Intangible Cultural Heritage by China in 2006 and by UNESCO in 2009 [6]. Its "discovery" by Chinese researchers in the 1950s as a polyphonic tradition and its subsequent promotion to battle older Western claims that Chinese music was entirely monophonic played an important role in the monumental changes in the musical practices of Kam villagers in the next sixty years years [7].

This section discusses issues of presence and displacement the contemporary practice of Kam Grand Choir, based on research on Kam music since the 1950s and our fieldwork in Kam villages between 2017 and 2023.

2.1 Historical Overview

Several studies on Kam music have focused on the evolution of the Kam Grand Choir, which reveal that prior to its recognition as Intangible Cultural Heritage on both national and international levels, this song genre had already been a subject of discussion and promotion, particularly in association with ethnic cultural development. Influenced by the new Chinese state's ethnic policies and the goal to enhance the image of the nation's music [5, 8, 9], since the 1950s the Kam Grand Choir began to expand outside of its traditional village context and became adapted to different staged settings [10]. Despite this wave of initial promotion, the following decades saw the Great Leap Forward (1958 - 1960) when the singing tradition was virtually abandoned due to famines and the Cultural Revolution (1966 - 1976) when village singing was prohibited and the stage singing focused on pro-communist rearrangement of songs [5]. Although the village singing resumed with increased political freedom in 1980s, the disruption of the previous decades and the various major socioeconomic shifts in the 1990s meant that earlier modes of performance and transmission of Kam Grand Choir have been faced with disintegration. Following the Economic Reform of 1978, large numbers of young Kam villagers migrated to coastal cities for work opportunities, leading to an absence of young people in the village most of the year. In Xiaohuang, the Kam village where our main fieldwork took place, many of these young or middle-aged migrant Kam workers returned to

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

the village when cultural tourism became a viable source of livelihood after the 2000s, especially following the Kam Grand Choir's promotion to Intangible Cultural Heritage status. Those of them who grew up during the decades when the village singing tradition was suspended were then relearning the songs for tourist performances, often with influences from staged performances. As a form derived from the village singing tradition, due to its growing popularity since the 2000s, staged performance of the Kam Grand Choir began to affect significant changes to the village performances, ranging from its repertoire, aesthetics, and social nature [5].

2.2 The Influence of Staged Performance on Village Singing

First, traditional Kam Grand Choir repertoires are deeply rooted in the specific locales of distinct villages, with each piece's name often signifying its origin, thereby emphasizing the music's strong connection to its village context. In contrast, the derived format of staged performance transcends these local boundaries, adopting a pan-Kam style that lacks the same depth of context, as the repertoire often feature songs from various areas that have often gone through aesthetic processing by non-Kam music professionals, no longer tied directly to specific villages [11].

Second, compared to its traditional village context, where the Kam Grand Choir is sung in New Year celebrations not simply as a musical performance, but also an important social activity that functioned as a form of dialogue and exchange between different singing groups, the staged performance could happen at any time of the year, often in front of a listening non-Kam audience. Instead of facilitating cultural transmission within the ethnic group, the derived format broadens its appeal to attract broader audience, including tourists and individuals from various ethnic backgrounds [5].

Third, the derived format tends to exhibit both an increasing rhythmic and melodic complexity and a shift towards the professionalization of singing and performance. This evolution reflects broader cultural shifts, where the younger generation often no longer learns to sing the Kam Grand Choir through years of immersion in childhood singing groups taught by the village sang ga, also known in English as "song experts" or "song masters", but instead receives musical education in Mandarin at schools outside their native villages. When they return to their communities, they blend these newly acquired techniques and styles to the local music practices [5]. This sometimes leads to confusion and misalignment when people gather to sing the songs in the village style but cannot stay in sync due to the different ways they have been taught to sing the song outside of the village.

Thus, due to the increasing influence of staged performance and the displacement of village singing practice at the confluence of drastic socioeconomic, cultural, and political shifts in recent decades, "tradition" in the contemporary practice of Kam Grand Choir is no longer a stable concept.

3. THE PERCEPTUAL MODE IN VIRTUAL REALITY

Virtual reality today merges the experience of controlled isolation with the immersive sensation of being close to a digitally constructed reality, enabled by highly individualized apparatus. Although primarily considered within the realm of visual technologies, this apparatus signifies a broader historical shift of perception, nuancing the relationship between the observer and the observed. American art historian Jonathan Crary's works offer insights to understand this perceptual paradigm by tracing the development from the camera obscura's mode of mediated observation of the 17th and 18th century to the emergence of subjective perception in the 19th century.

3.1 The Perceptual Paradigm of Camera Obscura

Crary's work elucidates that during the 17th and 18th centuries, the camera obscura exemplified a vision paradigm characterized by a fixed interior-exterior relation between the observer and the observed [12]. Within the confines of the camera obscura, the observer became an isolated subject, trapped within a dark chamber that distinctly separated them from the external environment. The camera obscura lends credibility to the individual's perspective, yet the experience it provides is a carefully orchestrated depiction of the external environment.

3.2 The Paradigm of Subjective Perception

Although the camera obscura's perceptual mode persisted into the modern era, the 19th century witnessed the emergence of a new paradigm, influenced by extensive research on the human body and the rise of psychophysics. A defining characteristic of this period was the blurring of the line between interior and exterior, with the observer's body emerging as a pivotal site for the management, measurement, and production of reality [13]. This new perceptual paradigm highlighted a double-edged development in the perception field. It empowered the individual, affirming the sovereignty and autonomy of the body in the perceptual experience; yet, it also made the body more vulnerable to control and annexation by external agencies, facilitating a path toward the increasing standardization and regulation of perception [13].

3.3 The Dual Perceptual Modes in VR

It is within the convergence of these two perceptual paradigms that we attempt to locate the potentials and limitations of the medium of VR. Through Crary's framework, virtual reality can be seen as representing an integration of both perceptual paradigms. It maintains an isolated environment, mirroring the camera obscura's distinct separation between inside and outside, the observer and the observed. While providing a close proximity for the audience to observe the virtual world, the apparatuses of bulky goggles and the confinement of physical space also keep the audience at a small distance away from full immersion, highlighting a sense of displacement and leaving space for critical reflection. At the same time, VR also enables a

highly individualized and subjective experience through controlled apparatus, echoing the emergence of subjective perception in the 19th century, where the body became a site for the management and production of reality. While interactive VR experiences often emphasize the possibility of evoking empathy through presence and immersion and giving agency to the spectator [14], they also make the body especially vulnerable to manipulations or biases of the authors of the experiences. In our VR opera "Loom", we work with the various possibilities and contradictions inherent to the medium and create a space for a nuanced engagement with the oral tradition of Kam Grand Choir.

4. "LOOM": A POLYPHONIC OPERA IN VR

"Loom" is a new exploration of polyphonic opera in VR by the authors. The project began with our fieldwork in Xiaohuang in 2017 - 2018, a popular Kam village for tourists celebrated for their rich heritage of Kam Grand Choir [15]. An in-headset single-player 6DOF interactive VR experience, the work-in-progress version of "Loom" was presented at the Oregon Shakespeare Festival in November 2022 and the Sound and Color Symposium at the Park Avenue Armory in New York in January 2023. It transports the audience into Lemon's (one of the authors) poetic reconstruction and speculation on her encounter with the presence and displacement of Kam music tradition in the village, as a Han Chinese composer who was briefly immersed in the learning and experience of Kam Grand Choir.

4.1 Process Overview

Initially, we engaged in field research within Kam villages in Guizhou province, particularly in Xiaohuang, capturing the soundscape, recording footage, conducting interviews, and learning traditional singing techniques from village song masters between 2017 and 2023. The empirical data collected during this period informed the creation of our PCVR project. We transposed the visual and auditory elements from our archive into 3D models and environments, utilizing tools like Cinema 4D, Substance Painter, and Unity for the development process. This project, designed in Unity with SteamVR, offers an immersive experience accessible through devices like HTC VIVE, Oculus Rift, or Meta Quest 2. By employing 3D visualization, storytelling, and spatial sound, we crafted a VR environment rich in detail and depth. The development was based on a comprehensive approach, incorporating archival research, interviews, field recordings, photography, and video footage, which together provided a multifaceted foundation for the VR composition and narrative. Through this workflow, music and sound design derived from recorded sounds with techniques such as impulse response, were integrated as a core element, producing an immersive soundscape in the virtual realm that reflects on our engagement with the complexities of Kam musical practice.

4.2 Immersive Presentation of Social Context

Compared to music intended as auditory experiences or as audiovisual works on 2-dimensional surfaces, music presented through immersive media could provide more context on the sociocultural conditions and physical environment within which the music is produced, experienced, and transmitted, and thus allow for a more nuanced engagement with the music tradition. In the case of "Loom", we situate the audience in a CGI-modeled village centering around the dare low (wooden pagoda, also known as drum tower) [11], an important Kam site charged with multiple dimensions of time and place. Over the course of the experience, the audience sees a new drum tower gradually form from a few desolate wooden beams, while ghostly figures and ruptured village landscape emerge from a murky field. Echoing the haunted transformation in the visuals are processed recorded samples of Lemon's voice and field recordings captured in Xiaohuang, which are re-embodied in various forms of spatialization, some as if coming from the audience's own body, some as if speaking through the drum tower, the trees, and the cicadas.

Figure 1. Still of the fractured landscape in "Loom".

Traditionally the place where Kam Grand Choir singing took place during the lunar new year period, in Xiaohuang and other Kam villages, most drum towers were torn down during the Cultural Revolution due to the repression of folk practices [16]. The destruction of the drum towers marked the prohibition of village Grand Choir singing practice during this period. Although many drum towers were rebuilt in recent decades at the original sites, the disruption of the village singing practice created a chasm and generational divide in the community's relation to the village singing practice. Due to the drastic social, economic, and cultural shifts in the villages since the 90s, such as the influx of cultural tourism, the earlier village modes of Kam Grand Choir singing gave way to influences from staged performances, leading to changes ranging from repertoire, aesthetics, social functions, and modes of transmission [5]. The emergence of a new drum tower from ruins in "Loom" points towards the lingering effect of this collective cultural trauma in the Kam music tradition today, the palpable presence of a past that comes back and haunts the here and now.

Figure 2. Still of the wooden beams of the drum tower in "Loom".

4.3 The Mode of Speaking Nearby

"Loom" does not claim to represent the reality of Kam Grand Choir, but rather leans towards the mode of "speaking nearby", a term coined by Vietnamese filmmaker and composer Trinh T. Minh-ha in her 1982 film "Reassemblage" featuring the everyday village life in Senegal [17]. It could be understood as a reflexive mode of presenting ethnographic materials that attempts to avoid laying claim to authority, leaves the space of representation open and through distance, fragmentation, and abstraction evokes the peculiarity of the medium and the fraught nature of ethnographic work. With its immersive evocation of fractured landscape, re-embodied voices, and disjointed time and space, "Loom" employs a poetic mode of expression that similarly seeks to avoid claims of authority, leaves open the space of representation and interpretation, while drawing attention to the shared state of presence and displacement the authors observed underneath both the mode of perception in Virtual Reality and the contemporary Kam Grand Choir tradition.

5. CONCLUSIONS AND DISCUSSIONS

To be developed into a full-length opera in VR, the current in-progress version of "Loom" presents a new exploration at the intersection of Kam music traditions and virtual reality technology. By immersing users in a digitally reconstructed Kam village, the project offers an interactive and reflexive encounter with the cultural and historical nuances of the Kam Grand Choir. Rather than treating the medium of VR as a 3D snapshot of a stable object, the immersive experience contemplates on the fluidity and multidimensionality of the oral tradition.

Our exploration through "Loom" allows us to dissect and reflect upon the broader implications of using VR as a medium to convey complex cultural narratives. It raises pertinent discussions about the role of immersive technologies in documenting and interpreting oral traditions. While VR offers a compelling platform for an immersive re-encounter with cultural practices, it also prompts a critical examination of how these practices are represented, experienced, and understood in a digitized context. These are important cultural and technical challenges that we will be tackling in our ongoing development "Loom" in the coming years. As we continue to meld the realms of traditional art forms and cutting-edge technology, it is imperative to foster a dialogue that acknowledges the complexities, challenges, and opportunities presented by this confluence.

Acknowledgments

"Loom" was co-commissioned by Quills Fest of the Oregon Shakespeare Festival in 2022.

6. REFERENCES

[1] E. Roaro, "Cinema memories in 3d modeling and virtual reality storytelling: The odeon cinema in udine," *Alphaville: Journal of Film and Screen Media*, no. 21, pp. 113–130, 2021. [Online]. Available: https://doi.org/10.33178/alpha.21.07

[2] T. Pistola, S. Diplaris, C. Stentoumis, E. A. Stathopoulos, G. Loupas, T. Mandilaras, G. Kalantzis, I. Kalisperakis, A. Tellios, D. Zavraka, P. Koulali, V. Kriezi, V. Vraka, F. Venieri, S. Bacalis, S. Vrochidis, and I. Kompatsiaris, "Creating immersive experiences based on intangible cultural heritage," in *2021 IEEE International Conference on Intelligent Reality (ICIR)*, 2021, pp. 17–24.

[3] E. Roaro, "Narratives of the future: Immersive stories / indigenous roots," *Body, Space Technology*, vol. 21, no. 1, pp. 113–130, 2022. [Online]. Available: https://doi.org/10.16995/bst.7960

[4] W. Wang and L. Jiang, *Ethnic Identity of The Kam People in Contemporary China*. Routledge, 2022.

[5] C. Ingram, "Tradition and divergence in southwestern china: Kam big song singing in the village and on stage," *The Asia Pacific Journal of Anthropology*, vol. 13, no. 5, pp. 434–453, 2012.

[6] UNESCO, "Grand song of the dong ethnic group," https://ich.unesco.org/en/RL/grand-song-of-the-dong-ethnic-group-00202, 2009, accessed: 2024-03-10.

[7] C. Ingram, "A localized perspective on china's intangible cultural heritage: The case of kam big song," in *Proceedings of the 18th Biennial Conference of the Asian Studies Association of Australia*, Adelaide, 2010.

[8] J. A. V. Aalst, *Chinese Music*. Paragon Book Reprint Corp., 1964.

[9] Z. Fan, 中国多声部民歌概 *[An introduction to China's multi-part folk songs]*. Renmin Yinyue Chubanshe, 1998.

[10] H. Zheng, 州侗族音 *[Guizhou Dong Ethnic Music]*. Guizhou Renmin Chubanshe, 1985.

[11] C. Ingram, "The shifting strength of place in contemporary big song singing from southwestern china," in *Presence Through Sound: Music and Place in East Asia*. Routledge, 2020, pp. 14–26.

[12] J. Crary, *Techniques of the Observer: On Vision and Modernity in the Nineteenth Century*. MIT Press, 1992.

[13] ——, *Suspensions of Perception: Attention, Spectacle, And Modern Culture*. MIT Press, 2001.

[14] E. Bond, *Displacements of Experience: The Case of Immersion and Virtual Reality*. Cham: Springer International Publishing, 2020, pp. 673–685. [Online]. Available: https://doi.org/10.1007/978-3-030-47178-1_46

[15] Y. Song and M. Yuan, "Tourism and its impact on dong traditional music and life in xiaohuang," *Journal of Tourism and Cultural Change*, vol. 19, no. 2, pp. 200–215, 2021. [Online]. Available: https://doi.org/10.1080/14766825.2019.1707839

[16] C. Ingram, "Echoing the environment in kam big song," *Asian Studies Review*, vol. 35, no. 4, pp. 439–455, 2011. [Online]. Available: https://doi.org/10.1080/10357823.2011.628008

[17] E. BALSOM, "'there is no such thing as documentary': An interview with trinh t. minh-ha," https://www.frieze.com/article/there-no-such-thing-documentary-interview-trinh-t-minh-ha, 2018, accessed: 2024-03-10.

EXPLORING THE SPACE BETWEEN INSTRUMENT AND CONTROLLER

Eleonora OREGGIA[1]

[1]**Goldsmiths, University of London**, London, UK

ABSTRACT

Electromagnetic sensing systems allow for expressive interaction in electronic music, performance, and time-based media art. REBUS is a machine which emits and receives electromagnetic waves to form a sensing space where human movement and presence can be detected with previously unknown precision. The potential of the system is explored through a workshop and a study targeted to selected groups: whereas the workshop observed the approach of coders and musicians in proposing compositional methods that this electromagnetic field sensing instrument offers, the study inspected the behaviour of classical musicians, mainly trained on strings or keyboard instruments, vis-a-vis that of selected electronic musicians. The terrain between instrument and controller is also explored through a fruitful deception. The paper narrates the conceptual assumptions behind these experiences through an initial analysis and some reflections on the data collected.

1. ELECTROMAGNETIC INTERACTIONS

This research explores electromagnetic sensing systems for expressive interaction in electronic music, performance, and time-based media art [1]. The research has materialised in a series of experiments and the creation of a musical system for audiovisual performance composition using electromagnetic waves (Figure 1).

REBUS is a machine that creates a sensitive space where any subtle interaction is detected independently of external light or sound and where invisible affordances can be touched and manipulated with the hands and the body, almost as invisible strings. The peculiarity of REBUS does not lay in the sonic register but in the original way in which it is possible to manipulate its oscillation, in the musical potential of the vibratory motion of the pressure waves modulated through manipulating electromagnetic waves [2].

Following the creation of this state of the art system for electromagnetic interaction, a series of activities, including a workshop, a public presentation and a user study, were designed to further explore the potential of the system, ultimately mapping the space between musical instrument and controller and considering whether playing with waves can be a new way to make music, or the electronic *par excellence* way to interact with sound, and whether an open and programmable instrument like REBUS can display recognisable characteristics that make it a musical instrument.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

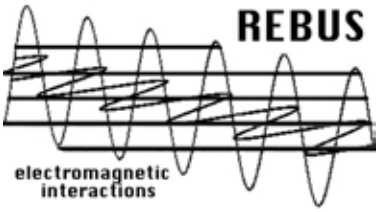


Figure 1. Electromagnetic Interactions logo and sign displaying an electromagnetic wave traversing a pentagram.

2. WHAT IS A MUSICAL INSTRUMENT

Starting from the assumption that "*...for a tool to be a musical instrument there must be at least a musician willing to play it...*" [2], the concept of musical instrument can be explored and clarified. According to Kvifte the concept of instrument cannot be disentangled from that of music and any reference to an instrument shall always be understood as part of a cultural context, innervated by economic interest and power relationships [3]. This view expands the understanding of musical instruments beyond the traditional loop *performer - bodily gestures - instrument - musical gestures*. An alternative approach explores the field beyond the limitations of the instrument-as-device evaluation framework, which, borrowing from traditional HCI, evaluates the device's success as a function of the tasks performed by a user-musician, limited by the characterisation of the musician as prototypical user [4]. A more conducive approach is suggested in the interpretation of the instrument as a constellation of processes or affordances which may mean different things to different musicians.

In this view the user is re-described as an ecology composed of agent (or agents) and environment, ultimately defining the coupling musician plus instrument as an interactive system. This approach is in line with the cultural perspective which explores musical instruments as the objects of material culture, which acquire a degree of power and agency in the context of their use: musical instruments are made to make meaning and they are therefore more than *the thing itself*, emerging as part of a continual convergence of meaningful developments in music, culture, design and technology, a relationship which should inform any attempt to answer the question "what is a musical instrument" [5]. Musical instruments as cul-

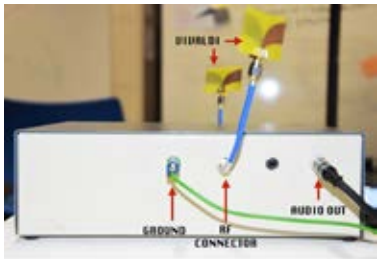


Figure 2. REBUS, first prototype, side view.

tural objects are therefore somewhat colonised by cultural habits of consumption and signification. In proposing a framework for the evaluation of digital musical instruments (DMIs), O'Modhrain proposes to include, along with players and audiences, also composers, instrument builders, customers and manufacturers, remembering that in many cases performers are also composers and eventually designers. This poses a challenge because both frameworks and taxonomies present limits, in particular when the evaluation is made from an audience's perspective, as most DMIs no longer present a perceivable or clear causal link between the gestures required to play the instrument and the mechanism that produces the sound [6]. According to this perspective the most important stakeholder in the process of designing and building a DMI is the performer/composer, although also audience, designer and manufacturer present perspectives that are all valuable at different phases in the design cycle.

In the study that follows, the framework is determined by the prominent presence of the coder, who composes and in some cases also plays the instrument.

3. IDENTITY

Following a long phase of engineering in order to construct the first prototype, the question whether REBUS is a musical instrument or a controller emerged as a consequence of two fundamental properties of the instrument: the form of interaction shapes the identity of the instrument; the sound palette, interaction logic and mappings offer infinite possibilities. REBUS is a machine that satisfies the definition identifying a musical instrument as formed by both interface and sound emitting engine. It is a small metal box with two protruding antennas (Figure 2), and its interface is created and analysed in realtime when the instrument is turned on [1]. The sound is generated by an embedded computer using the electromagnetic data to modulate the sound.

A controller, instead, would be attached to another, computer or device, and it would be used to *control* the other engine to which it is attached. However, the question was wider, and multiform: on the one hand, REBUS had been imagined, designed, enginered, programmed and played only by its author. This proved also the assumption that for an object to be a musical instrument, there should be someone ready to play it. But that is not enough to validate its ontological existence: would any other musician desire this particular instrument? And how would they use it? Can an instrument so ductile in terms of composition (any type of composition can be designed and programmed), but so determined in terms of interaction, be useful and attractive also for others who didn't invent it specifically to satisfy their performance needs? But more specifically, can this machine formally made *as seen in a dream* become a musical instrument with a defined identity and a mark that made it recognisable the way we could possibly identify an acoustic instrument by hearing its sound? In other words, is the identity of an instrument something that you can hear?

3.1 Coders as Agents

These questions provoked a series of activities to come to an understanding of the instrument that goes beyond an individual approach and experience. This phase of the Electromagnetic Interactions research project was targeted to coders, musicians and producers and included a workshop, a public presentation, and a user study to collate qualitative and quantitative data to answer the question whether REBUS can be considered a musical instrument and whether the electromagnetic field, when used as an interface to interact with sound, is not *innocent*, ie it fosters a deeper characterisation of the instrument itself as well as the sound as it is produced independently from the composition devised.

The workshop selected invited participants to write code for the instrument for one day. The target group included coders who are also musicians: some experience with Digital Signal Processing and C++ was a requisite. The group was composed of a combination of students and non-students aged roughly between 20 and 60 years old. Invited participants were divided in small groups, and each group was given a Bela, an integrated microprocessor specialised in real-time audio and interaction, along with the necessary instructions and the code skeleton to get started. Following a demonstration of the instrument and a brief introduction to coding with Bela/REBUS lasting one hour in total, each group had a day to produce an original composition, ie software that, uploaded on REBUS, defines the link between movement and sound (Figure 3). Throughout the day the workshop leader, also author of the instrument, assisted and troubleshot the various groups. The idea to consider workshops as a participatory process and creative practice which can provide more relevant insight than the classic completion of a task of the user-study paradigm is not new, as workshops have already been interpreted as a rich delivery instrument for emergent forms of music [7]. Adding coders to the social framework validates the idea that algorithms, and by extension the code that implements them, can be interpreted as score [8].

This occasion was the first public workshop where participants had the possibility to program and make their hands dirty with REBUS. At the end of the workshop all participants had to respond to the following three questions, the first requested them to comment on their experience ('*How was your experience with REBUS?*'); the second hinged on

[1] https://xname.cc/rebus

Figure 3. REBUS workshop at Goldsmiths, University of London, April 2023.

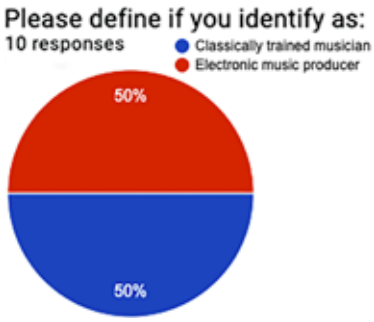


Figure 4. Pie-chart showing the identification of the target groups.

expectation ('*Was it different or similar to what you had imagined and expected?*) and the third explored the question of limits ('*What do you think are the limits of this form of interaction and music making?*').

Most users responded that the instrument was exactly as imagined, while one asserted it being "*...something you have to try...*" another said "*it's the opposite of a predictable experience*" and insisted in the third question that "*unpredictability could look like a limit*" but "*repeatability and predictability should not be part of the definition of a worthy musical experience*". Some participants expressed a certain enthusiasm, defining REBUS as '*a truly new and exciting experience*' and '*a maverick kind of making sounds*'. A participant declared that '*the creative dimensions yet to be explored with an instrument that harness something as nebulous as the electromagnetic spectrum are vast*'.

3.2 Musicians and electronic music producers

The user study targeted two main groups: classically trained musicians, and electronic musicians (non-classically trained). The two groups were not necessarily mutually exclusive, however the discremen hinged on the musician's own identification as either a classical musician or an electronic music producer (Figure 4).

The demographic of the sample subjects in terms of gender, age, ethnicity was sufficiently varied. Ten subjects were invited to test the instrument and comment on it. Participants completed a feedback form before the study (Intro), and another one following the study (Outro), the latter partially reported in Figure 6 and Figure 7.

They also received background information beforehand, but no instruction regarding the compositions or explanation regarding how to play was offered. The attempt was

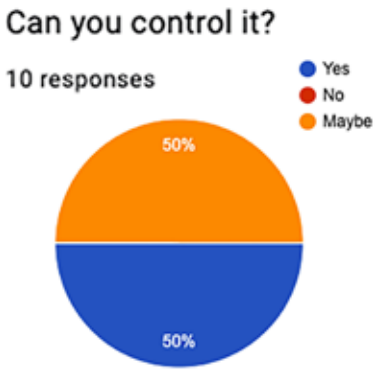


Figure 5. Half of the study participants are unsure whether controllability is a property of a musical instrument.

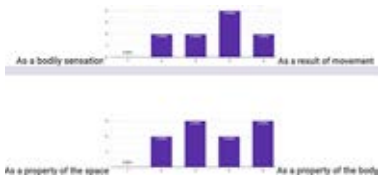


Figure 6. *How did you perceive your interaction with the sound?* As a bodily sensation - As a result of movement. As a property of the space - As a property of the body. Likert-scale answers, Outro form.

investigating whether selected target groups would display different approaches in interacting with REBUS, and whether they interpreted the machine as a controller or as a musical instrument. The participants, placed in front of the instrument in an immersive audiovisual performance space [2], were asked to explore and engage with four different compositions for three minutes each. When the three minutes passed they had to press the only button to move to the next composition.

The study presented a different set of questions, for example whether musicians trained on instruments that are more difficult to play and where the sounds have to be found by subtle adjustments of the hands, as for example the violin, would find REBUS more playable than those who were used to discrete and repeatable instruments such as the piano, where the action-response mechanism tends to be linear (pressing a certain key produces a certain note). Additionally, a number of electronic and noise musicians were invited to participate, in the attempt to find a correlation between habit, expectation and perception of the instrument.

An interesting fortuitous deception was also implemented to explore the element of illusion: how much of our interpretation is an illusion of the senses and perception is modulated by approximation and adjustment to expectation. The first six participants used REBUS with a loose antenna, decreasing its sensitivity of 80% circa.

At the end of the study the participants commented on the different compositions through a questionnaire, explaining how they perceived the results of their interaction, whether they could achieve a form of musical control which would

[2] Sonic Immersive Media Lab at Goldsmiths, University of London.

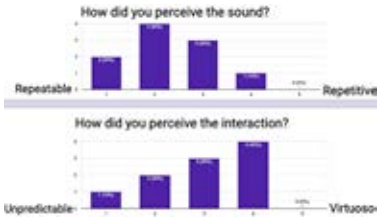


Figure 7. *How did you perceive the sound?* Repeatable - Repetitive. *How did you perceive the interaction?* Unpredictable - Virtuoso. Likert-scale answers, Outro form.

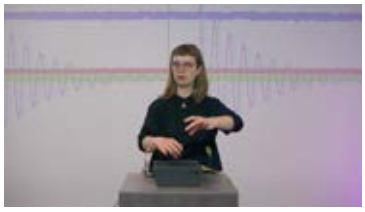


Figure 8. REBUS user study, task 2. Goldsmiths, University of London, May 2023.

respond to their aesthetics, and if they enjoyed playing the instrument. They were filmed while they explored the four compositions (Figure 8, Figure 9 and Figure 10), which were each addressing a different topic; for example, the fourth composition, a drum machine featuring invisible buttons to activate different features, was designed to be almost impossible to control, whereas the second, created by Claude Heiland-Allen, one of the participants to the workshop, was controllable but not predictable at the same time. Composition one was instead designed to generate sounds in a stochastic manner, with user's movement affecting waveforms alterations, while composition three explored complex interaction with simple timbre, mimicking an arpeggiator. All compositions shared the same mapping of the electromagnetic data.

4. OBSERVATION

What emerged from the ethnographic observation of the first study was that, among classical musicians, those who used to play string instrument, like the viola or the violin, appeared quicker in understanding a totally different form of interaction from what they expected. Their approach appeared more experimental, using both hands and ears as feedback and only partially using the visual feedback. The first six users showed signs of frustration and made wide use of the visual feedback that was projected on the three walls surrounding the performance space, with the exception of the violin player who displayed an incredible capability of understanding the interaction using fingers almost as if they had a brain of their own. The violinist managed to produce rhythmic patterns also with the first composition which was designed to be very little interactive and aimed at triggering frustration. They also found the point of silence in the second composition which is the most difficult

Figure 9. REBUS user study, task 3. Goldsmiths, University of London, May 2023.

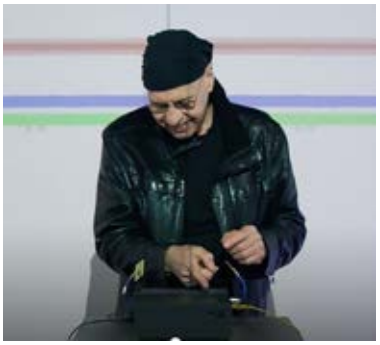


Figure 10. REBUS user study, user 10 task 4. Goldsmiths, University of London, May 2023.

to explore, and only seemed unsure at the fourth composition and its lack of linearity (the hidden buttons move along with the waves). This was even more surprising given the user tried the instrument with the deception on, but did not show any signs of frustration.

The noise musician, who tested the instrument without deception, approached REBUS slowly and, using fingers and both hands, displayed a greater capability to find borderline places, and spaces where the instability of the instrument could become a compositional strategy. They only appeared hesitant with the fourth composition (Rhythmbus), because of the initial lack of sound while the player was still searching for the invisible buttons.

5. ANALYSIS

Both the workshop's feedback forms, the two study's surveys and the video recorded tasks were passed through reflexive thematic analysis, which started with the familiarisation with the data to progress to the coding process. The process was both *inductive* and *deductive*, on the one hand aimed at giving voice to the data and the participants minimising bias and expectations, on the other remembering the research question underlying the experiment and the overall motivation at conducting it. The coding process was repeated twice, analysing the data in different orders and positioning the researcher's perspective as a component or reflexive aspect of the analysis [9].

Following a first analysis of the audiovisual and textual material collected, the semantic and latent themes emerging were that of 'Experience', which grouped codes describing the quality of the subjective experience such as

'pleasant', 'exciting', 'frustration'; 'Interaction', which grouped codes exploring the interrelation between subject and object, and 'Ontology', which grouped adjectives attributed to the instrument, such as 'novelty', 'freedom', 'subtle', 'complex', 'dynamic'.

From the answers to the final question,"*Define your performance with REBUS*", which therefore implied a sort of relationship between the subject and the instrument, a series of words appeared to point towards an almost sentimental metaphor; where the pianist said '*I wasn't very satisfied*' with my performance, the viola player asserted '*It was a getting to know between me and the instrument*' and '*I experienced a lot of self-reflection*'. The violin player, on the other hand, despite exploring the composition with decreased sensitivity, declared that the instrument '*has a not so difficult learning curve, very enjoyable*' and '*it feels natural to produce sounds with hand movements*'. User 4, an electronic music producer, said instead that he felt like '*balancing a chaotic system*', something that '*might take time to learn*'. User 5 and User 6, electronic music producers with an interest in novel musical instruments, both performing with the decreased sensitivity, defined the system respectively as '*mysterious and challenging but rewarding*' and '*an exploration of a relationship with something uncontrollable*'. The following users, electronic and noise musicians, described it as '*a hands on experience with a subliminal plane*', with one asserting '*it gave me the desire to continue exploring*', and '*super fun*'. Not only the discrepancy of experiences was surprising, but it almost appeared as if this physical interaction with something perceivable (there was an action-response effect) but impalpable at the same time triggered some forms of emotional response towards the *electromagnetic entity* the subjects were relating to.

6. REFLECTIONS

As the invisibility gives space for the expectation of unpredictability, we used a little deception and conducted the study with the first six participants without fully tightening the receiving antenna, therefore greatly diminishing the sensitivity of the system. Additionally, the four compositions implemented diverse techniques displaying varying reactivity to the interaction, for example the fourth composition was in purpose very difficult to control, if not almost uncontrollable. The idea to introduce deception responded to the need to reduce bias in the participants' answers, offering an extra margin of confrontation and verification of the data collected. Additional themes may include 'Level', where the discussion oscillates between the oppositions difficult and easy, and 'Pleasure', where the quality of the experience is discussed in hedonistic terms.

7. FUTURE

The themes and codes abstracted can model ongoing observation on real word interaction with the instrument through longer workshops culminating in performance opportunities where the users have access to the instrument for an extended time. The space into which practice increases mastery, performers work with composers, and Theremin players make music with waves, will be explored in future research phases over diverse demographics.

The activities described in this paper form the methodological and theoretical backbone of a conceptual grid which will be purported and structured in future experiences. The initial analysis described and the first set of reflections around the workshop already suggested the possibility that REBUS is not only a sound producing device or a controller or a musical instrument, it might rather be a proto instrument (the Theremin is different in that it uses the electric field) proposing a form of interaction whose physical mechanics in the void, translated into acoustic pressure, are so peculiarly identifiable that it may constitute a novel family of musical instruments, exploiting not strings or brass or woodwinds but electromagnetic waves.

Acknowledgments

The study described in this paper was made possible by Early Career Research Fund, Goldsmiths University of London. The authors would like to thank Rebecca Superbo, Sebastian Gafencu and Niccolò Perego from Polytechnic of Milan (POLIMI) for their assistance during the workshop and for contributing code and ideas for the study. Last but not least, gratitude to Professor Augusto Sarti and Professor Mark D'Inverno for believing in our work.

8. REFERENCES

[1] E. Oreggia, "The EM field as a sensor system," 2013.

[2] ——, "Audiovisual performance composition using electromagnetic waves," Ph.D. dissertation, Queen Mary, University of London, 2020.

[3] T. Kvifte, "What is a musical instrument," *Svensk tidskrift för musikforskning*, vol. 90, no. 1, pp. 45–56, 2008.

[4] M. Rodger, P. Stapleton, M. Van Walstijn, M. Ortiz, and L. Pardue, "What makes a good musical instrument? a matter of processes, ecologies and specificities," in *NIME Proceedings*. Birmingham City University, Birmingham, UK, 2020, pp. 405–410.

[5] K. Dawe, "The cultural study of musical instruments," in *The Cultural Study of Music*. Routledge, 2012, pp. 195–205.

[6] S. O'Modhrain, "A framework for the evaluation of digital musical instruments," *Computer Music Journal*, vol. 35, no. 1, pp. 28–42, 2011.

[7] K. Jo, A. Parkinson, and A. Tanaka, "Workshopping participation in music," *Organised Sound*, vol. 18, no. 3, pp. 282–291, 2013.

[8] T. Magnusson, "Algorithms as scores: Coding live music," *Leonardo Music Journal*, pp. 19–23, 2011.

[9] V. Braun and V. Clarke, *Thematic analisis. A practical guide*. Sage, 2022.

PAPER & POSTER /DEMO SESSION 8

/ Balancing Physical Modeling and Musical Requirements: Algorithmically Simulating the Calls of Hyalessa Maculaticollis for Real-time Instrumental Control

/ A Universal Tool for Generating Datasets from Audio Effects

/ Stochastic Resonance: Molding Sounds from Noise

/ Speculative Machine Learning in Sound Synthesis

/ Unveiling the Timbre Landscape: a Layered Analysis of Tenor Saxophone in RAVE Models

/ Tempo Estimation on Multiple Metrical Levels with Sequency Flux and Multiresolution Analysis

/ Correlations Between Objective and Subjective Evaluations of Music Source Separation

/ Reconstructing the Charlie Parker Omnibook Using an Audio-to-score Automatic Transcription Pipeline

/ Exploring Deep Learning Methodologies for Music Emotion Recognition

/ ChordSync: Conformer-Based Alignment of Chord Annotations to Music Audio

/ Prevalence Of Tresillo Rhythm In Contemporary Popular Music

BALANCING PHYSICAL MODELING AND MUSICAL REQUIREMENTS: ALGORITHMICALLY SIMULATING THE CALLS OF HYALESSA MACULATICOLLIS FOR REAL-TIME INSTRUMENTAL CONTROL

Staas DE JONG (staas@freedom.nl) [1]

[1] **Universiteit Leiden**, *Honours Academy*, Leiden, NL

ABSTRACT

This paper presents an algorithm that simulates the calls of the Hyalessa maculaticollis cicada for musical use. Written in SuperCollider, its input parameters enable real-time control of the insect call phase, loudness, and perceived musical pitch. To this end, the anatomical mechanics of the tymbal muscles, tymbal apodeme, tymbal ribs, tymbal plate, abdominal air sac, tympana, and opercula are physically modeled. This also includes decoherence, following the hypothesis that it, in H. maculaticollis, might explain the change in timbre apparent during the final phase of a call sequence. Overall, the algorithm seems to illustrate three main points regarding the trade-offs encountered when modeling bioacoustics for tonal use: that it may be necessary to prioritize musical requirements over realistic physical modeling at many stages of design and implementation; that the resulting adjustments may revolve around having physical modeling perceptually yield sonic events that are well-pitched, single-attack, single-source, and timbrally expressive; that the pitch-adjusted simulation of resonating bodies may fail musically precisely when it succeeds physically, by inducing the perception of different sound sources for different pitches. Audio examples are included, and the source code is structured and documented so as to support the further development of cicada bioacoustics for musical use.

1. INTRODUCTION

Reflecting on immersive sound, music and computing – the theme of this year's conference – one way in which forms of instrumental control can be designed so as to recognizably tie musical pieces in which they are used to specific geographical regions, time periods, or circumstances, is to have the heard sounds that they produce mimick those that occur in the corresponding, natural environments.

Simultaneously, however, another musical requirement may well be that these heard sounds also should have a clearly perceived pitch which can be changed in real time – e.g. to enable melodies; and that across such pitch or other control changes, the resulting heard sound will still, via the process of human Auditory Scene Analysis (ASA, see Bregman in [1]), be perceived as coming from a single source – e.g. to help enable use in counterpoint.

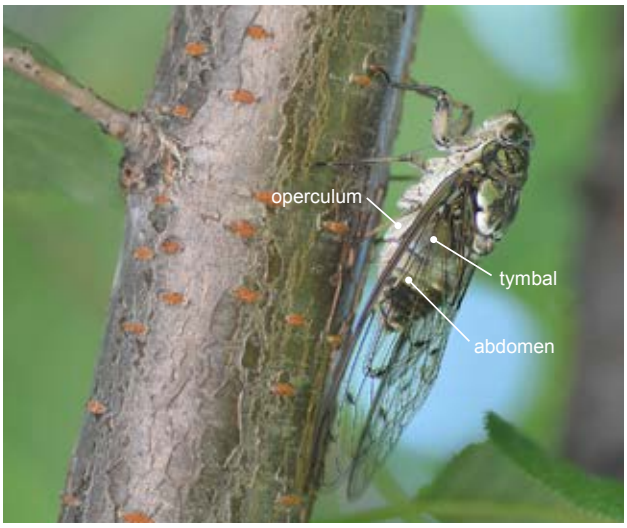


Figure 1. An H. maculaticollis individual in its natural environment, showing its tymbal, abdomen & operculum. [2]

This paper presents an algorithm that attempts to balance these three musical requirements for the case of H. maculaticollis, a species of cicada that is native to the Korean Peninsula.

Admittedly, when looking for a suitable starting point to investigate physical modeling versus musical requirements, choosing tonal use may not immediately seem like an obvious choice. After all, are there not more recently introduced musical paradigms available, subject to more dynamic ongoing developments?

However, if we do not want to first have to debate the contents of a specific musical paradigm, and instead want to focus on its interplay with physical modeling, this seeming disadvantage becomes an advantage: The requirements for tonal use are relatively well-understood, static, and non-controversial. For example, it seems relatively clear-cut that tonal aesthetics will usually be violated if sound synthesis does not provide musical voices with accurately and precisely controlled perceived pitches.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[2] Cropped and annotated version of a photograph by 더 그레이스, licensed under https://creativecommons.org/licenses/by-nc/4.0/ and obtained via [2].

1.1 Approach: Physical Modeling of Anatomical Mechanics, Informed by the Analysis of Field Recordings

Earlier work on synthesizing cicada sound includes [3] by Georgaki and Queiroz. Here, a different species, Cicada orni, is studied, which produces calls that in their timbre and rhythm are quite different from those made by H. maculaticollis. Still, this work clearly illustrates how, generally speaking, cicada sound synthesis can be based on the analysis of field audio recordings. For the present work, two field recordings in particular, kindly made publicly available at [4] and [5], have been used as a reference throughout development.

However, a choice was made to also base development on biological knowledge of anatomical mechanics, motivated by the potential of H. maculaticollis to provide a detailed case study of where along the causal structure of bioacoustics trade-offs between realistic modeling and musical applicability may lie.

For gaining the required biological knowledge, Pollack provided a very useful starting point in [6], giving a clear introductory overview of the anatomical components and mechanics of cicada sound production, based on earlier work by others, while placing this within a wider context of insect bioacoustics.

H. maculaticollis can then be approached as a specific instance of cicadas and their sound production, with some of the key anatomical components illustrated in Figure 1. Here, an individual of the species is shown residing in its natural environment, with the anatomical locations of its left tymbal, abdomen, and left operculum indicated. The tymbal is an organ containing ribs that can be sequentially pulled inward by the tymbal muscles, via the internal connecting tissue of the tymbal apodeme. The resulting reverberations go into other parts of the tymbal; into the air sac that is contained within the abdomen; and into the surroundings, especially via the left and right tympana. Here, each tympanum, while itself an acoustically open membrane, is variably covered by its corresponding operculum.

In order to start, based on the above general context, on the development of a concrete algorithm, the information provided by Bennet-Clark and Young in [7] and [8] has been crucial: Detailed description and measurements regarding the relevant anatomy and its mechanics and acoustics are provided, as well as a rigorously tested model of the resulting sound generation, including abdominal resonation. Although this is based on the study of other species of cicada, especially Cyclochila australasiae, results are evaluated as a model for sound production in cicadas in general, and seemed a suitable starting point when attempting to develop the algorithmic simulation of cicada calls by H. maculaticollis.

Further on, the work by Smyth and Smith reported in [9] and [10] was helpful: As a part of reported work on a novel musical instrument, two general digital filter types are mentioned that can be used when implementing the modeling of C. australasiae tymbal and abdominal resonation, respectively.

2. DESIGN AND IMPLEMENTATION OF THE ALGORITHM

The algorithm was written from scratch in SuperCollider, and is included in annotated source format in the Appendices. It has six input arguments, each of which can be changed in real time: `a_gate_bit`, `loudness_nrm`, `t_mae_trig`, `t_mae_retrig`, `t_mi_trig`, and `freq_hz`.

The function of audio-rate argument `a_gate_bit` is to enable graceful termination of the algorithm. This both in terms of the amplitudes of its stereo audio output, and in terms of the release of claimed computational resources. Via the control-rate argument `loudness_nrm`, the perceived loudness of algorithm output can be set, along a scale from 0.0 to 1.0.

Unlike C. australasiae, perceptually, the calls of H. maculaticollis seem to follow a three-part structure, consisting of an initial phase, a repeated middle phase, and a final phase. The arguments `t_mae_trig`, `t_mae_retrig`, and `t_mi_trig` make each of these phases, respectively, subject to real-time control: A 1.0 impulse to any of these trigger-type arguments signals the algorithm to transition to the corresponding phase of the call. Here, parts of the transliterated onomatopoeic Korean word for cicada, "maemi", are used as a mnemonic, to indicate which phase of the insect call each argument refers to. For an example of the use of these input arguments to control a complete call sequence, see the waveform at the bottom of Figure 3.

Finally, the remaining, control-rate argument `freq_hz` is used to relay, as a frequency in hertz, the intended perceived musical pitch that algorithm output should induce in the listener. Here, we encounter the first trade-off between realistic physical modeling and musical requirements: Cicada song frequencies are emphatically present at thousands of hertz, but the modeled input range for `freq_hz` was set to the octave from E4 to E5. This so that the heard pitches would still lie within one of the main ranges of tonal music: that of the soprano, the highest human vocal range.

The subsections below will give an introduction to the algorithm itself. For reasons of clarity, this will be done by traversing various parts of the insect's sound-generating anatomy in a causal order. The information on cicada anatomy and sound production that is used will be based on [7] and [8], unless explicitly stated otherwise.

2.1 The Tymbal Muscles: Contraction & Relaxation

During cicada song, the tymbal muscles, connected to the left and right tymbal, go through repeated cycles of contracting inward and then relaxing. In the algorithm, this was modeled in the `tymbalMuscle_trig` signal, a unipolar waveform consisting of single-sample impulses, one for each contraction cycle. During the segments of cicada song where the tymbal ribs buckle in tight synchronicity (and thereby produce a tightly coherent pulse train), each `tymbalMuscle_trig` impulse models the moment in time when contraction causes the first tymbal rib to start to buckle.

Moving on to computing the *frequency* of muscle contractions during such segments, the available information on C. australasiae presented a clear problem: Live control of the algorithm should, as discussed above, yield perceived musical pitches lying between 329 and 660 Hz. This, however, is both above the 100-200 Hz range of natural tymbal muscle contractions, and below the resulting natural song frequency of around 4300 Hz.

To resolve this, a musically-motivated choice was made to have each contraction always cause exactly 4 buckling events (see Section 2.3 for details on their nature). The idea was that, in this way, muscle contraction modeled at a mostly natural rate could cause strong overtones at the precise pitch frequency specified, with the insect song overtones then present above that (while not being perceptually dominant as the pitch). Accordingly, the starting point for the computation of the `tymbalMuscleFreq_hz` signal was a division by 4 of `freq_hz`.

Beyond this, computing `tymbalMuscleFreq_hz` was based on analysis of the H. maculaticollis field recordings. Here, a first observation was that during the initial and middle phases of ordinary call sequences, tymbal muscle frequency seemed to be at mostly higher levels associated with tightly synchronous tymbal rib buckling, to then end at lower levels during the final phase. For the algorithm, a musical choice was then made to model these lower levels so as to be a whole number of octaves lower. The idea was that this might help harmonically: First, to clearly connect the final call phase to the initially heard pitch; and second, to avoid unintended dissonances during use with other musical voices in a piece. A subsequent choice, returning to primarily considering physical modeling, was then to go a single whole octave lower, but not more, as the latter seemed to yield results sounding too similar to a general end of cicada calling.

After this, a more involved physical modeling choice was to simulate the control of tymbal muscle frequency in further detail by implementing a system of envelopes. Here, each call phase was modeled by a separate envelope, with each envelope starting or terminating based on the live input from `t_mae_trig`, `t_mae_retrig`, and `t_mi_trig`; and with all envelope output combining into a single, continuous signal.

Within the individual envelopes, the target levels, durations, and curvatures defining each envelope segment were – except when based on the octave difference mentioned above – directly derived from measurements of features present in the field recording obtained from [4]. Where possible, this was done by comparing a set of 12 adjacent harmonics that were evident in spectrograms sampled at 11025 Hz. These were visible in the 3.3-5.3 kHz range of the field recording, and in the 0.9-2.7 kHz range of algorithm output (when testing at pitch E5). The results of these measurements have been explicitly labeled and unambiguously structured in the algorithm, using a series of 15 scalar constants that define the envelopes, and 4 signal variables that contain their output.

While doing the above, a musical choice was made to *not* mimick a general downward trend seen spanning across the harmonics of repeated middle call phases in the field recording. This to help avoid algorithm output going out of tune during tonal use.

A final, speculative addition to tymbal muscle frequency control was then to give an explicit presence to the idea of it also being subject to further, possibly unknown but smaller additive effects. This was done by raising or lowering the computed frequency by a semi-normally distributed random value, updated at least once every tymbal muscle contraction. The peak amplitude for this was limited to 10 cents, however, as going higher resulted in ripples in muscle frequency harmonics that were not visible in the field recordings. Perceptually, the result then seemed to yield at most a slight variation. In any case, this extra computation can be effectively removed from the algorithm by changing a single line of code.

Example output of the live computation of `tymbalMuscleFreq_hz` during a full call sequence at pitch E5 can be seen in the waveform placed second-lowest in Figure 3. The corresponding `tymbalMuscle_trig` signal can be heard in of the Appendices.

2.2 The Tymbal Apodeme: Pull & Release

Each muscle contraction cycle, the tymbal apodeme, being the connecting tissue between a tymbal muscle and the tymbal itself, is pulled inward with variable force, and then released. Relevant results from the literature on C. australasiae here include that for tymbal rib buckling events, their phase coherence as a pulse train is supported by the mechanics of the insect's anatomy from tymbal apodeme to abdominal air sac; and that at higher apodeme pulling speeds, it appears *"that the tymbal buckling mechanism offers good coupling between the vibrations of the ribs so that the rapidly excited tymbal inevitably produces a coherent waveform"*.

This information influenced physical modeling in the algorithm via two main points of interpretation: First, that we will assume that apodeme pulling speed is related to tymbal muscle contraction frequency. As a first approximation for modeling this, in the algorithm, the first stage of computing the `tymbalApodeme_pullingSpeed_nrm` signal has been to make it directly proportional to the output of the tymbal muscle frequency envelopes.

The second point of interpretation was then that, above a certain threshold of apodeme pulling speed, tymbal mechanics will pull buckling into (the same) tight coherence. In the algorithm, this was implemented by modeling `tymbalApodeme_pullingSpeed_nrm` as a normalized value, and then clipping and scaling it to a maximum corresponding to the threshold. The concrete value for the threshold was then chosen so that `tymbalApodeme_pullingSpeed_nrm` would mostly be at its maximum level during the initial and middle phases of H. maculaticollis call sequences – the parts where in the field recordings, tymbal muscle contraction frequency mostly seemed to be at higher levels associated with tightly synchronous

tymbal rib buckling. The resulting computation of `tymbalApodeme_pullingSpeed_nrm` is illustrated by the waveform placed third-lowest in Figure 3.

2.3 The Tymbal Ribs: In-Buckling & Out-Buckling

2.3.1 Modeling the Individual Tymbal Rib Buckling Events

The literature indicates that for C. australasiae, during normal song, a single pull-and-release cycle by the tymbal muscle and apodeme results in the first 2, sometimes 3 tymbal ribs sequentially buckling inward, from posterior to anterior (while the 4th rib only buckles during "protest song"); and that after this, there is a simultaneous buckling back outward.

For the algorithm, as a trade-off between physical modeling and musical requirements, the choice was made to always model 4 buckling events per muscle contraction (this has been motivated in Section 2.1): the in-buckling of tymbal ribs 1-3, followed by their combined out-buckling. The associated acoustic buckling pulses were then modeled using one envelope per buckling event, with each event represented by a bipolar, single-pulse waveform. The amplitude levels, durations and curvatures of the envelope segments defining these waveforms were based on the acoustic measurements presented in [7] in Figures 3, 4, and 7, and Table 2.

During normal song by C. australasiae, there are 2 or 3 whole cycles of the tymbal plate resonant wave (see Section 2.4) in between the buckling pulses (see e.g. Figure 5a in [7]). In the algorithm, the overall time between the start of consecutive buckling events already was to be fixed as the reciprocal of the currently intended musical pitch, as indicated via the `freq_hz` input signal. Within this context, the duration of each buckling pulse was then scaled so as to allow for precisely 2 in-phase echoes before the start of the next pulse.

Regarding the pulses' amplitudes, the overall loudness of tymbal rib buckling events may be subject to further anatomical control: In [11], it is noted for other cicada species that contraction of a tensor muscle *"increases the convexity of the tymbal and therefore the power required to make it click"*. This then causes the tymbal ribs to produce a louder sound once they do buckle. In the algorithm, the effect of this is mimicked by linear amplitude scaling, which was made subject to live (musical) control via the lower half of the input range of `loudness_nrm`.

Figure 2 shows an example of a complete sequence of tightly synchronized tymbal rib buckling pulses as computed for pitch E5 by the algorithm.

2.3.2 Modeling Decoherence

So far, we have already recapitulated a crucial part of how it is that the anatomy of a singing cicada can transform individual tymbal rib clicks into a longer, continuously oscillating pressure wave: The tymbal mechanics of muscle contraction, apodeme pulling, rib buckling, and plate reverberation cause a series of acoustic pulses that are tightly in phase with eachother. By "decoherence", then, we will here mean the loss of this phase coherence.

For the algorithm, we will follow the hypothesis that in H. maculaticollis, decoherence might explain the change in timbre that is apparent during the final part of a call sequence. In spectrograms of the field recordings (for an example, see the top of Figure 3) it is visible how, during the initial and middle call phases, the sound intensities at frequencies around insect song frequency are distributed across separate harmonics; while during the final phase, these intensities are still relatively higher than elsewhere in the spectrum, but much more evenly distributed. This can, of course, also be heard.

The literature indicates that, for C. australasiae, a loss of phase coherence is associated with slower apodeme pulling speed resulting in greater intervals between rib buckling events [7]. For the algorithm, it was assumed that a similar association would exist for H. maculaticollis. As a first approximation of concretely modeling this, the duration of intervals between rib buckling events was made inversely proportional to apodeme pulling speed.

However, the proportional relationships modeled so far – i.e. between tymbal muscle contraction frequency, apodeme pulling speed, and rib buckling intervals – would then not suffice, still yielding clearly separate harmonics during the final call phase. Therefore, as a first approximation of modeling the effects of further unknown processes, at each tymbal muscle contraction, the pre-buckling interval duration added by decoherence was drawn, according to a uniformly random distribution, from none up to the proportionally computed amount.

An example of the resulting algorithm output for pitch E5 can be compared to one of the field recordings in the rightmost part of the spectrograms at the top of Figure 3, starting vertically above the `mi_trig` impulse in the graph at the bottom. Also, the underlying, internally computed `tymbalRibs_buckling_sig` signal can be heard in of the Appendices.

2.4 The Tymbal Plate: Resonant Vibrations

As is described in the literature on C. australasiae, when a tymbal rib buckles, this also rapidly moves inward another part of the tymbal organ: the tymbal plate. This plate is also connected to the dorsal pad, another component of the tymbal made of a rubber-like protein called resilin. The overall mechanical result is that a buckling event will set the tymbal plate into a series of resonant vibrations, which then become damped over time. Acoustically, this means that each buckling pulse will cause a longer waveform, which then decays approximately exponentially. Two properties can then be used to characterize such a waveform: the frequency of its wave cycles, and their "quality factor" Q. The latter is a scalar measure that quantifies the decrease over time of successive peak amplitudes.

To complicate matters, however, each buckling event modifies the actually resonating mass, and thereby also the frequency and Q value of tymbal plate resonation.

In the algorithm, separate instances of a resonant bandpass filter were used to model the different resonations following each type of buckling pulse. Here, the four center

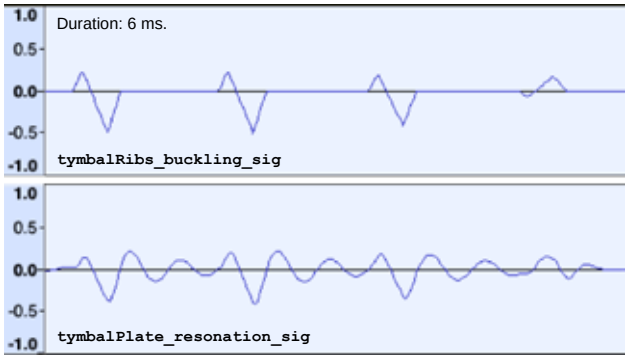


Figure 2. *Top:* example of simulating one complete sequence of tightly synchronized tymbal rib buckling events (at pitch E5). *Bottom:* the corresponding simulated response of the tymbal plate.

frequencies were explicitly computed from the four mean resonant wave frequencies presented for C. australasiae in [7]; while being scaled to 3× `freq_hz`, so that two decaying wave cycles would fit in between tightly coherent buckling pulses, regardless of the currently required musical pitch. The four corresponding Q values for C. australasiae (measured using acoustically unloaded tymbals) were then also used explicitly, to compute the different bandwidths of each filter. Here, care was taken to first halve the Q values, so as to make them representative of the acoustically loaded tymbals of an intact cicada.

The overall `tymbalPlate_resonation_sig` signal was then computed as the weighted sum of the signal containing the simulated buckling pulses and those containing the tymbal plate after-resonations. Here, the weights were set so that the resulting waveform was similar to the acoustically measured results from literature. This included the property that the decayed peak amplitude of the resonant wave cycle occurring directly before the next in-buckling pulse should still be above 25% relative to the maximum – acoustically representing a mechanical vibration amplitude still powerful enough to help start the next tymbal rib's in-buckling (thereby supporting coherence).

The results of this are illustrated in Figure 2, which shows an example tymbal plate resonation waveform caused by a complete sequence of tightly synchronized tymbal rib buckling events, as computed by the algorithm. An example `tymbalPlate_resonation_sig` signal, internally computed for a full cicada call sequence at pitch E5, can be heard in of the Appendices.

Finally, one speculative addition was made to the computation of tymbal plate resonation. This was based on the impression that perceiving increased loudness during the calls of H. maculaticollis seemed to possibly involve not only an increase in overall amplitude, but also a relative increase of the intensities at higher frequencies. This seemed evident, for example, when listening to the field recording at [5], and then taking two 4-second snippets from its first and second half, amplifying these to the same peak amplitude, and comparing the spectrograms, especially in the 5.0-8.0 kHz range.

This motivated the idea to extend the algorithm with the simulation of some mechanism that would cause its output, too, to include a relative increase at higher frequencies when increasing perceived loudness levels. This seemed desirable not only from a physical modeling perspective, but also from a musical one: The suspected phenomenon seemed reminiscent of how increasing the attack force while playing notes on acoustic musical instruments can have an expressive impact by not just increasing the amplitude, but also the brightness of the tones heard.

The mechanism then concretely modeled was itself highly speculative: Its starting point was that, beyond some level of increasing applied tymbal muscle force, the maximum displacement of the vibrating tymbal plate may become limited to some fixed amount, due to its physical attachment to the rest of the cicada's body. Acoustically, this then *might* result in the wave peaks of tymbal plate resonation becoming progressively flattened as vibration amplitude increases, thereby yielding more intense overtones at higher frequencies. In the algorithm's `tymbalPlate_clippingResonation_sig` signal, the effects of the computations implementing this tentative idea can be observed. However, these effects can also be avoided, easily and completely, by not using the upper half of the `loudness_nrm` input range. Alternatively, the extra computations can be removed from the source code by using an also-included single-line alternative.

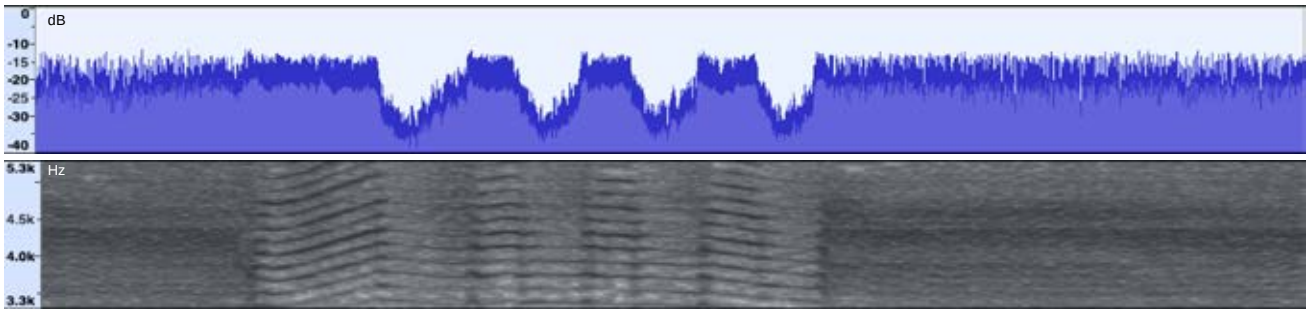
2.5 The Abdominal Air Sac, Tympana & Opercula: Helmholtz Resonation

As is described in the literature on C. australasiae, movement of the tymbal plate causes air pressure changes going into the abdominal air sac. Together with the left and right tympana – which, variably covered by the opercula, act as the effective sound sources radiating into the external surroundings – this air sac forms an abdominal resonator that appears to produce bottle-shaped or Helmholtz resonation, having the effect of a narrow band-pass filter. Here, the dominant resonant frequency f_0 is closely tuned to that of the tymbal, and can be modeled by the equations used in [8] that are listed in (1).

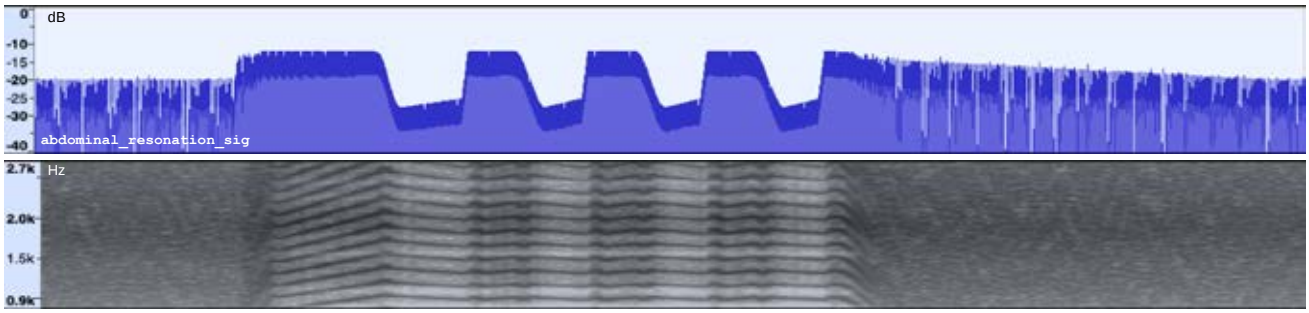
$$
\begin{aligned}
f_0 &= \frac{c}{2\pi} \cdot \sqrt{\frac{A}{L \cdot V}} \\
A &= 2 \cdot a \\
L &= 16 \cdot r \,/\, 3\pi \\
r &= \sqrt{a \,/\, \pi}
\end{aligned}
\tag{1}
$$

Here, c is the speed of sound; A is the area of the neck, being twice the area a of a single tympanum; L is the *effective* length of the neck, determined by a via the single-tympanum equivalent hole radius r; and V is the volume of the cavity. From these dependencies, it then follows that the main variables remaining to be modeled for the case of H. maculaticollis are V and A, i.e. abdominal volume and effective tympanal area.

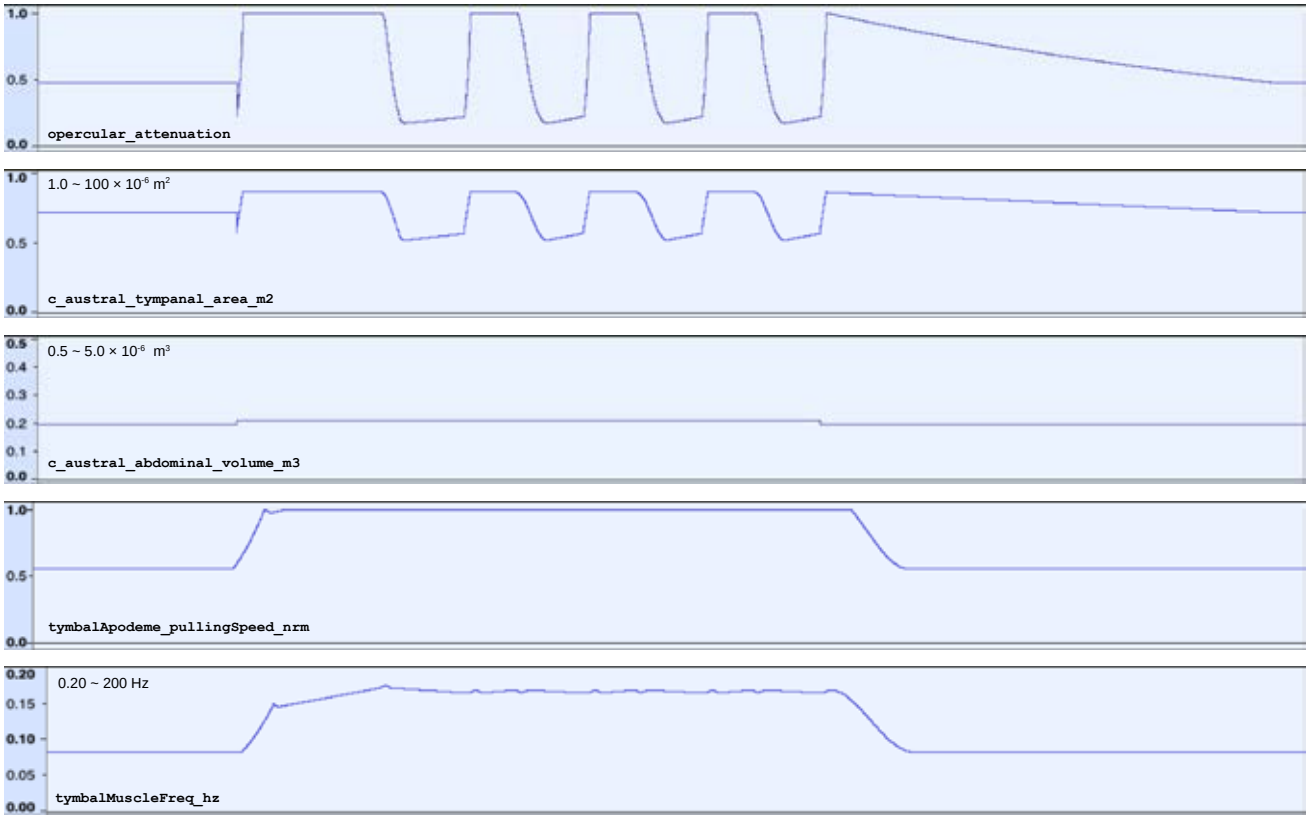
Waveform & spectrogram: field recording of a complete H. maculaticollis call sequence. *Duration everywhere: 6.6 seconds.*



Waveform & spectrogram: example of corresponding algorithm output computed for pitch E5.



Waveforms of selected, corresponding internal signals:



Waveform of corresponding real-time input signals:

Figure 3. Algorithmically simulating the calls of H. maculaticollis. Each signal is further explained in the main text.

In the algorithm, abdominal resonation was modeled using a 6-pole (i.e. -36 dB/octave) resonant bandpass filter. Its bandwidth was tuned so as to make algorithm output harmonically similar to H. maculaticollis field recordings, using spectrum plot measurements of the progressive drop in amplitude of the harmonic peaks below insect song frequency (details are in the source code). Fortunately, musically, there was no actual trade-off here when testing at pitch E5, as the resulting bandwidth value seemed to fall within a middle range only outside of which the unintended perception of pitches E6/B6 and E4 seemed to start to occur.

Then, for computing the abdominal volume over time, the starting point were measurements of unextended and extended abdominal cavity volume for C. australasiae, derived from [8]. To this was then added the general idea that H. maculaticollis' abdominal cavity might be less extended during its final call phase. Once modeled, the effects of this seemed hard to discern, however, both via auditory perception and in spectrograms, and the interested reader is therefore referred to the source code for further motivation and details. The resulting computation of `c_austral_abdominal_volume_m3` is illustrated by the waveform placed fourth-lowest in Figure 3.

In comparison, modeling H. maculaticollis' effective tympanal area over time was considerably more involved, once similarly started from the overall tympanal area reported for C. australasiae in [8]. In general, modeling was based on the observation that both cicada species raise their abdomen in some way while singing, thereby lifting the tympana away from the opercula which cover them; [3] and also, on the observation in spectrograms of the H. maculaticollis field recordings that during the initial and middle call phases, this modulation of effective tympanal area seemed to be reliably synchronized with specific patterns in tymbal muscle contraction frequency.

As for tymbal muscle frequency in Section 2.1, a system of envelopes was implemented, one for each insect call phase, with the specific target levels, durations and curvatures defined by a total of 26 scalar constants explicitly based on measurements of amplitude and spectral features present in the field recording obtained from [4]. During this, at three points in the algorithm, musical requirements were consciously prioritized over realistic physical modeling. In each case, envelope segment properties were adjusted so as to perceptually make each simulated insect call phase better correspond to the perceived attack of a single tonal event. This affected 6 constants. [4] The details of all of this are documented in the source code.

A fourth point where musical requirements were prioritized over realistic physical modeling was when recomputing the abdominal resonation frequency resulting from the above relative to the currently intended musical pitch (as specified via `freq_hz`). This, however, will be discussed in the Discussion section.

The resulting computation of `c_austral_tympanal_area_m2` is illustrated by the waveform placed fifth-lowest in Figure 3. Also, an example of the overall resulting `unattenuated_abdominal_resonation_sig` signal can be heard in of the Appendices, as it was internally computed for a full cicada call sequence at pitch E5.

2.6 The Opercula: Amplitude Attenuation

In the literature on C. australasiae, a reduction in sound pressure level is reported when fully closing the opercula. Accordingly, in the algorithm, too, amplitude attenuation of the abdominal resonation signal was implemented, based on the current effective tympanal area. This was done down to a dB SPL level set so that algorithm output matched the drop in strongest harmonic peak amplitude observed during middle call phases in the H. maculaticollis field recording at [4]. An example of the resulting `opercular_attenuation` signal is illustrated by the waveform placed sixth-lowest in Figure 3, as it was internally computed for a full cicada call sequence at pitch E5. The corresponding `abdominal_resonation_sig` signal can be visually compared to the field recording using the spectrograms and waveforms aligned at the top of Figure 3, while it can be heard in of the Appendices.

3. DISCUSSION

From the literature on C. australasiae, it is clear that its abdominal resonation – when abdominal volume is fully extended, and tympanal area fully uncovered – is strongest around the same frequency as tymbal plate resonation. Accordingly, in the algorithm, the abdominal resonation frequency initially was scaled so as to match tymbal plate resonation frequency. This meant that it, too, followed the currently intended musical pitch.

However, when testing this at pitch E4 instead of E5, output from abdominal resonation perceptually seemed to change not only in pitch, but also in timbre, and this to such an extent as to suggest origination from another, different sound source. This seemed problematic, in hindering algorithm output from being perceived as coming from a single source, e.g. during melodic use with other musical voices. An example of this pitch-following abdominal resonation can be heard in of the Appendices, which contains the `abdominal_resonation_sig` signal for a full call sequence at pitch E4.

Scaling abdominal resonation frequency relative to a *fixed* pitch E5 then seemed to perform better in this regard. In , the corresponding example of fixed-pitch abdominal resonation can be heard. Both options are preserved in the source code.

4. CONCLUSION

In Sections 2 and 3, we have discussed the design and implementation of a new algorithm that can simulate the calls

[3] For an H. maculaticollis video example, see: `https://youtu.be/XFOaGPNT55U?t=157`.

[4] `c_MAETRIG_TYMPAREA_START_NRM`, `c_MAETRIG_TYMPAREA_RISE_DUR_S`, `c_MAERETRIG_TYMPAREA_START_NRM`, `c_MAERETRIG_TYMPAREA_RISE_DUR_S`, `c_MITRIG_TYMPAREA_MAX_NRM`, `c_MITRIG_TYMPAREA_END_NRM`.

of H. maculaticollis, subject to real-time control. Now, revisiting this, as a case study of the possible trade-offs when modeling bioacoustics for tonal use, we can identify 11 instances, extending from the input parameters to the pre-final abdominal resonation stage, where a musical requirement was consciously prioritized over faithful physical modeling.

This was done for the soprano `freq_hz` input range; the pitch-inducing number of buckling events per muscle contraction; the whole-octave drop in tymbal muscle frequency; the harmonics kept in tune across repeated middle call phases; the pitch-adjusted tymbal plate resonation frequency; the optionally increasing timbral brightness during increasing loudness; the pitch-respecting abdominal resonation bandwidth; the three tympanal area envelope details for better matching insect call phases with single-attack tonal events; and the fixed-pitch abdominal resonation frequency.

This seems to illustrate a number of points that may be relevant when modeling bioacoustics for tonal use. First, that it may be necessary to prioritize musical requirements over faithful physical modeling at many stages of design and implementation. Second, that the resulting adjustments may revolve around having physical modeling perceptually yield sonic events that are well-pitched, single-attack, single-source, and timbrally expressive. Third, that an inherent contradiction may be encountered (here, while modeling abdominal resonation, see Section 3): The pitch-adjusted simulation of resonating bodies may produce sonic results that fail musically precisely when they succeed physically, i.e. when inducing the perception of different sound sources at different pitches.

Finally, the algorithm itself is provided in the Appendices of this paper, together with the audio illustrations that have already been used to demonstrate the computation of some of its key internal signals. In addition to this, demonstrates the tonal use of overall algorithm output together with another instrument, containing brief examples of counterpoint and chordal accompaniment by a digital piano. Throughout the algorithm's source code, emphasis has been put on readability and ease of reuse: The overall structure is flat and unambiguous, and uses human-readable constants and variables together with detailed in-line comments that explicitly motivate each of the choices made. The intention here was to make it easier for (in principle) anybody to use, modify or extend the algorithm. This could include changes to more faithfully simulate the calls of cicadas in general (e.g., by explicitly modeling a *pair* of tymbals); changes to better match H. maculaticollis, C. australasiae, or any other species of cicada; and changes to the trade-offs between realistic physical modeling and musical use.

Acknowledgments

I would like to take the opportunity to thank my hosts in Gyeongju for their sincere hospitality. Visiting this region formed a great part of my first (and hopefully not last) stay in the Republic of Korea, during which I first heard the calls of H. maculaticollis in real life.

5. APPENDICES

The appendices are included inside the PDF of this paper. Each embedded file can be opened or exported using its (shortened) name in the text. For the Audio Illustrations, this is via: , , , , , , , and . For the source code of the algorithm, this is via: . The latter needs to be placed among the Extensions of the SuperCollider installation that is then used to run the code (details provided). Finally, example source code that instantiates and parametrizes the algorithm is provided via: .

6. REFERENCES

[1] A. Bregman, *Auditory scene analysis: The perceptual organization of sound*. The MIT Press, 1990.

[2] iNaturalist contributors, "Hyalessa maculaticollis occurrence dataset," iNaturalist.org research-grade observations via https://www.gbif.org/occurrence/2563524032, 2018, accessed on 2023-12-28.

[3] A. Georgaki and M. Queiroz, "Virtual Tettix: Cicadas' sound analysis and modeling at Plato's Academy," in *Proc. Int. Conf. Sound and Music Computing*, Maynooth, Co. Kildare, Ireland, 2015.

[4] V. Ciubotaru, "Singing_cicada_audio.ogg," file at Wikispecies via https://species.wikimedia.org/wiki/File:Singing_cicada_audio.ogg, 2014, accessed on 2023-12-29.

[5] Σ64, "Hyalessa_maculaticollis_v01.ogg," file at Wikimedia Commons via https://commons.wikimedia.org/wiki/File:Hyalessa_maculaticollis_v01.ogg, 2012, accessed on 2023-12-29.

[6] G. Pollack, "Insect bioacoustics," in *Acoustics Today*, vol. 13, no. 2. Acoustical Society of America, 2017.

[7] H. Bennet-Clark, "Tymbal mechanics and the control of song frequency in the cicada Cyclochila australasiae," in *J. Experimental Biology*, vol. 200, 1997.

[8] H. Bennet-Clark and D. Young, "A model of the mechanism of sound production in cicadas," in *J. Experimental Biology*, vol. 173, 1992.

[9] T. Smyth and J. Smith III, "Applications of bioacoustics in physical modeling and the creation of new musical instruments," in *Proc. Int. Symp. on Musical Acoustics*, Perugia, Italy, 2001a.

[10] ——, "A musical instrument based on a bioacoustic model of a cicada," in *Proc. Int. Computer Music Conf.*, Havana, Cuba, 2001b.

[11] A. Ewing, *Arthropod bioacoustics: Neurobiology and behaviour*. Comstock Publishing Associates, 1989.

A Universal Tool for Generating Datasets from Audio Effects

Stefano Fasciani (stefano.fasciani@imv.uio.no) [1],
Riccardo Simionato (riccardo.simionato@imv.uio.no) [1], and
Aleksander Tidemann (aleksander.tidemann@imv.uio.no) [1]

[1] *Department of Musicology*, **University of Oslo**, Norway

ABSTRACT

This paper presents an open-source software tool that simplifies and automates the generation of accurate and reproducible datasets from audio effect devices. The tool is universal as it is designed to work with both software and hardware devices, presenting analog or digital interfaces. Specifically, this tool facilitates the process of recording the processed sound while varying effect control parameters, with four different paradigms offered. The tool has an interactive graphical user interface that facilitates interfacing audio effects and configuring dataset generation options. The generated dataset is organized as a collection of sample-accurate files, which makes further processing or analysis with offline scripts simpler. The paper details the tool's design and its extensive set of functionalities, which are informed by research in machine learning modeling of audio effects. While discussing the tool's applications, we also explore its potential in other contexts.

1. INTRODUCTION

1.1 Audio Effects

Audio effects, also known as effect units or effect processors, are electronic devices that alter the sound of an audio source using analog or digital signal processing techniques. These are widely used in music production, music performance, and broadcasting. They comprise of distortions, dynamic, filter, modulation, pitch, and time effects, although hybrid or combined effects are also common. These effects can be standalone devices or integrated within sound synthesizers, mixing boards, amplifiers, or performance-oriented systems. They can be implemented using analog circuitry, digital signal processors, or in some old instances, mechanical components. Standalone hardware effects vary in form factor, including pedals, tabletop units, and rack-mountable devices. The advent of standardized audio plugin formats in the 1990s led to an increase in software implementations of digital audio effects. These process the sound using techniques available exclusively in the digital domain, such as finite impulse response filters, or are designed to emulate classic analog effects. This approach, called virtual analog [1], emerged in the 1990s, and a variety of numerical techniques have been proposed since then, to replicate the nonlinear and time-variant characteristics of these devices.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Audio effects present standardized interfaces for input and output audio signals. Most effects process either one (mono) or two (stereo) audio channels, while larger numbers of audio channels are found in spatialization and surround processors. On hardware devices, the audio interface uses standardized unbalanced or balanced connectors, such as RCA jacks, 3.5mm or 6.35mm TRS or TS phone jacks, or XLRs. These are used to transport AC-coupled, line-level variable voltage analog signals. Some devices may also feature microphone or instrument level inputs. In software devices, uncompressed digital audio is generally exchanged through buffers of channel-interleaved audio samples, where the buffer size depends on the system's audio drivers. Each audio sample is represented as a 16- or 24-bit integer or as a 32- or 64-bit floating point number.

Audio effects typically provide control parameters, enabling users to adjust and customize the sound-altering process. These parameters influence specific, and often perceptually relevant aspects of the audio processing chain. In hardware devices, control parameters are mostly interfaced to variable resistors in the analog circuitry, while, in digital devices, they determine values of variables within the digital signal processing algorithms. Audio effects vary significantly in their control parameters. Besides the considerable variability in their number, the interfaces also differ. In hardware devices, these parameters are usually adjusted through dedicated or shared (i.e., re-mappable) knobs, sliders, switches, and buttons. Alternatively or additionally, they may be controlled also through analog Control Voltage/Gate (CV/Gate) interfaces or through a Musical Instrument Digital Interface (MIDI), with the latter being more widespread. MIDI-controlled parameters, using the common 1.0 standard, are indexed with a 7-bit integer and have a limited 7-bit resolution. The 2.0 MIDI standard, introduced in 2020, overcomes this limitation, but it's yet only available on a limited number of systems. CV/Gate offers theoretically infinite resolution, though the signals can be generated or received by a digital interface with finite resolution, typically in the range of 10 to 16 bits. Software effect units often use the Virtual Studio Technology (VST) or Audio Units (AU) formats. In these cases, all control parameters are indexed with an integer and represented by floating-point numbers in the range of 0.0 to 1.0.

1.2 Datasets of Audio Effects

A dataset generated from an audio effect typically includes output recordings for different values of the control parameters. As audio effects don't generate audio signals, the dataset is intrinsically linked to the selected input signal. The input signal selection largely depends on the specific application for which the dataset is developed, though it commonly includes a variety of stimuli such as sinusoidal tones, sine sweeps, amplitude sweeps, and combinations of noises. Excerpts of various music genres, recordings of individual percussive and non-percussive musical instruments, and singing and spoken voices are also frequently included. Copyright of the selected stimuli determines the extent to which the resulting dataset can be used and released. The output for the selected input signal is recorded for all combinations of the control parameters to be included in the dataset. However, this is not a strict requirement, and input of different durations and content can be used. The number of parameter combinations can vary significantly depending on how many variable parameters and their possible values are included in the dataset. While the term "permutation" might be more mathematically appropriate to identify unique settings of the control parameters, we use "combination" as we formally consider the value of each individual parameter belonging to a different set. Indeed, even after normalization, each parameter included in the dataset may have a different range and sampled values. There is no standardized way to arrange the dataset. However, recent efforts have aimed to minimize the likelihood that dataset differences can lead to significantly different experimental results. This has led to the development of an open-source library that handles datasets in their current distribution modes while regulating possible variability [2]. Common practices in developing an audio effect dataset include: providing a separate uncompressed audio file for each parameter combination with an identical sampling rate, and typically associating filenames to metadata in a Comma Separated Value (CSV) file. Encoding metadata in the filename is a less common option. The sampling rate can also vary significantly across datasets, with 44.1, 48, and 96 KHz being the most common choices. The selected stimuli input file is also usually included in the dataset. Datasets may also be arranged as a single audio file for the recording. In this case, boundaries for segmenting the recorded audio according to different combinations of parameters must be provided.

1.3 Applications and Uses of Datasets

Datasets of audio effects have various applications, and the specific purpose can influence the content and structure of the dataset. One use for these datasets is for modeling an audio effect using data-driven methods such as black- or gray-box approaches. Deep learning techniques have become increasingly popular in this context in recent years. Datasets can also be used to evaluate the accuracy of numerical models, even when these models are developed with explicit white-box techniques that do not rely on the device's data. In addition, these datasets can be utilized for offline precision characterization of an audio effect, measuring level, gain, noise, harmonic and intermodulation distortion, relative phase, crosstalk, and transfer function. Datasets can be used to compare different effect units that are designed to apply similar alterations to audio signals, such as those from different manufacturers, production batches, or after exposure to varied operating conditions or life cycles. Intelligent mixing systems, an emerging field, is another context where audio effect datasets can find significant applications. For example, datasets can be used in developing algorithms for recognizing applied audio effects and estimating parameters of an effect that generates processed signals. Datasets can be used to generate a semantic or perceptually-relevant model of an effect's response to parameter changes, assisting in the navigation, selection, and generation of parameter presets for a specific unit. In a creative context, these datasets can represent a rich and varied sonic corpus due to the significant alterations that audio effects can produce on the sonic content of an audio signal. Such data can be used as material for granular or concatenative synthesis, for music production, or artistic installations.

1.4 Challenges in Generating Datasets

There are several challenges associated with the generation of audio effect datasets. Firstly, it can be a time-consuming and tedious process. Assume three variable parameters are included in the dataset, with 10 sampled values from each, resulting in 1000 combinations. If we record 60 seconds of audio for each combination, it requires 16.6 hours of continuous recording time. Depending on the application, datasets may require a significantly higher number of sampled values for each parameter. While the total recording time might not be a significant issue, changing settings and restarting the recording could potentially introduce errors if executed manually. Consistency across recordings is critical- parameter settings, recording levels, and synchronization of playback and recording must be maintained, sometimes down to sample-accurate levels. If the setup must be dismantled and reassembled daily, there may be accuracy issues due to inconsistencies in the setup, particularly concerning settings controlled by non-digital means, such as gains in the audio interface or in the effect unit. These inconsistencies and errors can be detrimental to specific applications that use the dataset. The results generated by designs using the dataset may present flaws or lack precision, eventually traced back to the dataset itself after lengthy inspections. Automating the dataset generation process can save time and reduce the chances of errors and inconsistencies. However, automation needs to consider the specific interface of the selected audio effect and requires software development and validation that could consume the saved time.

The tool we propose in this paper addresses these challenges, offering fully automated and accurate dataset generation that is compatible with most hardware and software audio effects. It provides a high degree of accuracy and exposes an extensive set of functionalities to users through a comprehensive Graphical User Interface (GUI). The design and implementation, detailed in the follow-

ing sections, are informed by our research in modeling analog audio effects using deep learning techniques, an application that requires highly accurate datasets, as explained later in the paper. The tool, named as Dataset Generator for Musical Devices (DGMD) is available as open-source software at `https://github.com/stefanofasciani/DGMD`. In this paper, we present the DGMD version focused on audio effects, while a separate version focusing on sound synthesizers is still under development.

2. RELATED WORKS

Variable control parameters are found in both audio effects and sound synthesizers. Managing these during dataset generation and handling the potentially large number of combinations pose a common challenge. Furthermore, effects and synthesizers often share similar interfaces and formats. Automatic and offline rendering of software sound synthesizers is a widespread practice in research modeling different aspects of these algorithms. Commonly, a basic synthesis algorithm is explicitly programmed to generate such a dataset. However, software sound synthesizers in VST format have also been used, providing a more realistic and complex study case. Additionally, software sound synthesizers in the VST format have been used, offering more realistic and complex study cases. In this context, RenderMan [1] —a command-line VST instrument host with Python bindings, written in C++—represents a highly effective and efficient option for dataset generation. RenderMan not only facilitates dataset creation but also provides analysis of the rendered audio. It has been utilized in [3] and in [4] for generating datasets to train systems for the automatic programming of sound synthesizers.

The Timbre Space Analyzer and Mapper [5]-a previous work by the first author—part of the conceptual inspiration behind the DGMD tool-is a mixed MATLAB and Cycling '74 Max system. It allows for the analysis and modelling of a sound synthesizer's timbre response, as well as the control of synthesis parameters from various representations of the timbre space. To accomplish this, the system internally computes a dataset from a VST sound synthesizer subject to variations of selected control parameters.

The dataset used to train real-time neural network models of guitar amplifiers, presented in [6], is generated from a hardware analog amplifier. The amplifier's parameters are controllable only through manually operated knobs. To ensure precision, consistency, and repeatability—challenges hard to overcome with a human operator—the authors have developed a robotic control system for the parameter knobs, consisting of an array of electric motors. Another notable aspect of this work is their sampling approach for control parameters. Instead of using a fixed grid, which may lead to overfitting, they employ a random sampling strategy. Parameters are drawn from a uniform distribution for each combination added to the dataset, ensuring an unbiased sampling of the control space throughout the dataset.

3. DESIGN AND IMPLEMENTATION

3.1 Objectives and Choices

The tool we propose for generating datasets from audio effects has been designed to achieve the following objectives:

- It should support any audio effects, ranging from software plug-ins to hardware effects with MIDI, CV/Gate, or manual interfaces.
- It should minimize manual intervention by automating the entire dataset generation process.
- It should be cross-platform, easy to install, and user-friendly, with all functionalities accessible through a single GUI.
- It should support the generation of sample-level accurate and reproducible datasets.

For the implementation, we have selected Cycling '74 Max [2] because it offers the best trade-off between the objectives listed above. It interfaces with audio plug-ins through a native object, while supporting a variety of audio drivers that can cover virtually any sound card needed to interface an external audio effect. Max allows the building of standalone cross-platform applications with complex GUIs, solving any problems related to operating system-specific system libraries, graphic libraries, audio, and MIDI drivers, as well as compiling and building the application. Furthermore, Max is a reasonable choice for ensuring the longevity of our tool. This environment has been updated and maintained for over 30 years and can handle any changes in operating systems that will inevitably occur in the future.

The time required to generate a dataset from software plugins can be significantly reduced using Max's non-realtime audio driver, which renders audio at the maximum rate allowed by the CPU computational power. While the same or better efficiency could potentially be achieved using VST hosts for text-based programming languages, we still see significant value in using Max. Notably, when using Max, the real-time mode can be used to open the plugin's GUI, select variable parameters, and fine-tune their range to be included in the dataset. This can be done while listening to the effect's output before recording the dataset, a crucial feature in this context. Then the non-realtime driver can be used to generate the dataset offline. In contrast, a fully script-based offline approach, aside from not allowing one to listen to, interact with, and view the plug-in's GUI, also requires advanced knowledge of the audio effect controls' range, name, or index. These could vary significantly from those documented in manuals or visible on the plug-in's GUI.

While Max offers precise signal processing objects for sending audio and control parameters to audio effects, as well as for recording the effects' outputs with sample-level consistent accuracy, it does have limitations in handling

[1] https://github.com/fedden/RenderMan

[2] https://cycling74.com/products/max

non-native data structures and in implementing algorithms that require iterative loops. To overcome this, we have coded a significant portion of the DGMD in JavaScript, which Max natively supports and integrates through dedicated objects. Utilizing JavaScript advantages the cross-platform aspect, although execution time and synchronization with the signal processing part are not guaranteed. Hence, we have limited all time-critical operations such as interfacing with the audio effect to Max's signal processing components. Non-time-critical tasks, such as file I/O operations, are generally implemented in a combination of JavaScript and Max objects operating at message rate.

3.2 Interfacing Audio Effects

The tool can send and receive audio from effects with one (mono) or two (stereo) channels while controlling up to ten different effect parameters. These are the limits of the current implementation, which can be easily extended. Software plugin effects are hosted directly inside the DGMD, with audio and parameters exchanged using the VST interface. Hardware effects are interfaced using an audio interface to send and receive audio signals. Most audio interfaces also feature a MIDI output, which can be used to send parameters to those effects with parameters digitally controlled via MIDI. Effects with a CV/Gate parameter interface instead, require a more specific type of audio interface with DC-coupled audio outputs, and one audio channel per controlled parameter. In Figure 1 we have illustrated how software and hardware effects are interfaced to the DGMD.

In the case of hardware effects that lack an analog or digital interface for the parameters, we suggest a MIDI-controlled electro-mechanical interface as an alternative to manual control. This type of interface could be constructed using various low-cost microcontroller boards that include a USB class-compliant MIDI interface. The microcontroller would receive MIDI control change messages used to set the angle of an array of 270 servo motors, each corresponding to a specific controlled effect parameter. The microcontroller should feature a number of Pulse Width Modulation (PWM) output pins at least equivalent to the number of parameters to be controlled. Consequently, the audio effect appears to the DGMD interface as if it includes MIDI-controllable parameters. Such hardware and firmware design for interfacing with most manual effects is relatively simple and generic, and we have included an example in the tool's repository. However, the physical connection between the servo motors and the effect unit is device-specific and must be custom-made, for example by 3D printing the required parts to hold the motors in place and to clamp onto the effect unit's knobs.

4. FUNCTIONALITY AND FEATURES

In this section we detail all functionality and features we have included in the DGMD tool, motivating their integration against specific applications. The GUI is visible in Figure 2, where yellow overlay letters are used to label sections serving different purposes. Detailed information

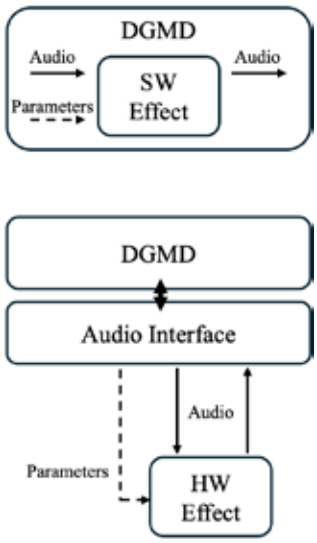


Figure 1. Interfacing between the DGMD tool and audio effects. Software (SW) plugin audio effects are directly hosted within the DGMD, with audio and parameters exchanged through the VST interface (top). In contrast, hardware audio effects are connected via an external audio interface, with parameters controlled through MIDI or CV/Gate signals (bottom).

about the functionalities is displayed to users when hovering the mouse over each interactive element of the GUI.

4.1 Device Selection and Interface

Section 'G' of the GUI allows users to specify the target audio effect device as either an internal software plugin or an external hardware unit. Users can select the number of audio channels for sending the stimulus to the effect and for recording the processed sound, which can be configured as mono or stereo. This section also lets users adjust and monitor the pre-recording gain of the audio signal from the effect and listen through the monitoring loudspeakers.

When working with internal plugins, section 'H' enables the selection of the specific plugin from system folders or the manual loading from a file. The plugin's GUI can be displayed in a separate window, and the 'Monitor Plugin' feature allows users to see the name and value of parameters changed in the plugin's GUI on the DGMD. This is critical because the DGMD tool displays the VST interface's parameters' names and values, which usually differ from those visible on the plugin's GUI. The VST names and values are necessary for determining what information to include in the dataset.

For external audio effects, the audio interface and selected channels for sending and receiving audio are chosen in section 'E'. If the effect has MIDI-controllable parameters, the output port and channel are set and enabled in 'F', and the corresponding MIDI control change numbers for each parameter are set in one of the ten 'D' sections. Additionally, CV control of parameters can be activated in 'F', with the associated specific DC-coupled output audio channel for each parameter defined in 'D'.

The dataset's recording sampling rate is determined in

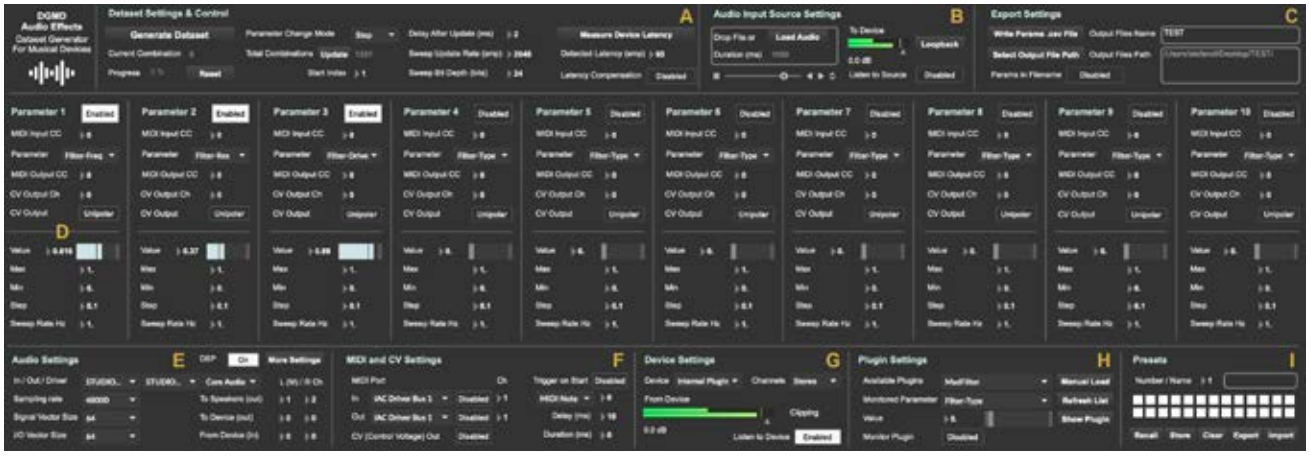


Figure 2. GUI of the Dataset Generator for Musical Devices (DGMD) tool, audio effects version. The capital letters in the yellow overlays indicate the different sections of the interface.

section 'E', where users can also choose the audio interface channels for the monitoring loudspeakers to listen to what is sent to and received from the audio effect.

We have included the option for users to manipulate the plugin or external effect's parameters via a MIDI controller, which is advantageous for testing parameters and ranges to include in the dataset and for setting parameters during manual dataset generation. Users can designate and enable the specific MIDI input port and channel for this purpose in 'D', and set the parameter-specific MIDI control change numbers in sections 'D'.

4.2 Parameters Options and Change Modes

The GUI of the DGMD, as shown in Figure 2, features ten identical sections labeled 'D', each corresponding to a specific effect parameter to be varied for dataset generation. A parameter value will only be changed by the tool during dataset generation if it is set to 'Enabled' in the GUI. To generate the dataset, at least one parameter must be enabled. Enabled parameters on the GUI don't necessarily have to be situated next to one another. For software plugin effects, a drop-down menu at the top of the 'D' sections allows users to select a parameter from the list of those available from the VST interface. For external hardware effects, within the same section, users can choose either the MIDI control change number or the audio channel for the CV signal. When selecting a CV signal, it is also possible to specify whether the controlled parameter is unipolar or bipolar, enabling the DGMD to map the parameter range to the appropriate output voltage values.

In the bottom portion of each section 'D', the current value of the parameter is shown both as a numerical value and as a horizontal slider. These indicators are interactive, allowing user adjustment and also changing automatically during dataset generation. Parameter values are consistently represented in the normalized range from 0.0 to 1.0. Additionally, users can select a specific range of interest-from 'Min' to 'Max'-for each parameter to be varied in the dataset, helping to reduce the time required for generation and to limit the overall size of the output files.

The dataset can be generated according to four different 'Parameter Change Mode', specified in the section 'A': *Step*, *Random*, *Manual*, or *Sweep*.

In *Step* mode, the DGMD generates the dataset by sampling the variable parameters using a regular grid. Each parameter is sampled using an independent 'Step' value, which users can specify in section 'D'. For instance, the settings displayed in Figure 2 show two enabled parameters, both with a selected range from 0.0 to 1.0. The first, however, has a step of 0.1, resulting in 11 different values, while the second has a step of 0.2, yielding 6 different values. As a result, and also as automatically indicated in section 'D' before dataset generation, the dataset will include 66 different parameter combinations. This provides users with an indication of how long the dataset generation will take, given the selected settings. While the current lowest step value is 10^{-5}, it is important to note that when parameters are controlled via MIDI, values are sent as 7-bit integers. In this case, appropriate step values are calculated as $1/2^N$, where N is the number of bits, ranging from 1 to 7, to consider. In devices with CV-controlled parameters, stages of quantization may also be present, especially when the signal processing is handled digitally. In these cases, CV analog signals may be acquired by ADCs with resolutions ranging from 10 to 16 bits.

Upon initiating dataset generation, the DGMD initializes a generator of parameter combinations. Within this generator, each combination is associated with a unique integer index. These indices are used progressively to query the generator, which then produces combinations on demand, one at a time, instead of storing them in a table—an approach that could consume a significant amount of memory. Subsequently, the parameter combinations are sent sequentially to the audio effect in ascending order according to their index. For each combination, the audio stimulus is sent to the audio effect, and the output is recorded. Audio data is initially read from and written into memory buffers to ensure sample-accurate synchronization. It is transferred to disk only after recording is complete, and before a new parameter combination is sent to the effect. A separate audio file is created for the output recording of

each parameter combination. Users have the option to set a delay time between updating the parameters on the audio effect and the start of playback for the input stimulus as well as the recording of the output. This delay accommodates any latency in the parameter update response time, which may vary from several milliseconds to fractions of a second, depending on the specific interface. In this case, an effect with electromechanically-controlled parameters represents the worst-case scenario.

Given that the dataset may include a large number of parameter combinations, potentially leading to a lengthy generation process, we have introduced a feature that allows users to begin generation from any selected parameter combination index. Coupled with the ability to save all settings in user-defined presets, the deterministic and reproducible behavior of the generator of parameter combinations, and the fact that each combination produces a separate audio file, users are empowered to create large datasets over multiple sessions without compromising the consistency or accuracy of the dataset.

In *Random* mode, the DGMD generates the dataset by drawing values from a series of independent uniform distributions, each corresponding to a different parameter. These distributions are confined to the selected range for each parameter, and in this context, the step value serves as a quantizer. This can be particularly useful when parameters are controlled using a low-resolution interface. For instance, with a range from 0.0 to 1.0 and a step of 0.01, only 101 distinct values can be drawn. Similar to step mode, in random mode parameter combinations are generated and sent to the audio effect on the fly, one at a time. However, in this case, the process is not reproducible, which is why the generated combinations are progressively stored in a memory table. Users need to specify the number of different random parameter combinations to be drawn to complete the dataset. The overall automated dataset generation process is the same as that described for step mode, including the option to delay the initiation of recording after dispatching a new set of parameters to the effect, and the ability to start from an index greater than one. However, the latter only affects the numbering of the recorded audio files, as a previously generated table including random combinations cannot be reconstructed. In such cases, users must manually merge the tables, which are exported to a file.

When the dataset is used to train a machine learning model of the audio effect, and the parameters are employed to condition the model's inference, sampling the parameters with a high-density uniform grid may lead to overfitting. In this scenario, random sampling emerges as a more effective strategy to mitigate overfitting. Nonetheless, for sparser grids — that is, those with fewer sampled values per parameter — a uniform sampling ensures better coverage of the effect parameter space. Still, a uniform grid, even if dense, might be the preferred choice for applications focused on analyzing, characterizing, or comparing the response of an audio effect.

The *Manual* mode enables users to set the parameters of the effect by hand. This mode is only semi-automated; each parameter combination is individually added to a memory table when users manually initiate the process of sending the stimulus to the audio effect and recording its output. This mode may serve various purposes: for example, it can be used to sample the parameter space using criteria different from those previously mentioned, likely involving only a limited number of combinations, or it can assist in the generation of datasets with external hardware audio effects with a manual interface for parameter adjustment. In such cases, users should duplicate the settings made on the effect unit within the DGMD interface before triggering the recording for each combination.

The *Sweep* presents major differences compared to other modes. Parameters are continuously varied while recording the dataset, each following a triangular wave with arbitrary range and frequency. In section 'D' of the GUI, it is possible to set the 'Sweep Rate', which is the frequency of the triangular oscillator controlling each individual parameter, which spans from the individual 'Min' to 'Max' defined in the same section. In this case, the concept of parameter combinations no longer holds, and the desired dataset size is expressed by the number of times the playback of the input stimulus signal is repeated. For each repetition, the effect output signal is recorded into an audio file. Also, the sweeping parameters are recorded at audio rate into separate multichannel audio files. As the recording processes are synchronized, each track on both resulting files presents an identical duration expressed in number of samples.

Control parameters for audio effects with a CV interface are transmitted as an audio signal. However, for all other interfaces, the triangular waves need to be sampled, and their values transmitted to the effect device. In section 'A' of the GUI, users can specify the 'Sweep Update Rate,' expressed as the number of audio samples, a feature effective only when the CV output is disabled. This determines the rate of a clock driving a sample-and-hold block, which downsamples the triangular waves and generates the actual set of values sent to the effect. Consequently, when the control parameters for the audio effects are recorded, the resultant triangular wave appears stepped. Moreover, to account for the eventual limited resolution of the interface controlling the parameters, in section 'A' users can select the bit depth for quantizing sweep values before recording.

We introduced the sweep mode for generating datasets that can be used to assess how audio effects respond to continuous variations of parameters. It can also serve to compare the behavior of an audio effect model versus the original device when parameters are varied continuously and gradually. Besides, a dataset wherein effect parameters are not fixed during the audio recording can be used for experimental training of machine learning models of audio effects. This allows to explore the extent to which such an approach might yield better results compared to models using a grid of fixed parameter combinations, which are required guessing outputs for unseen parameter values. The information embedded in the dataset and the potential ability of a model to learn from it can significantly vary with the selected rates. Slower variations may prove ad-

vantageous for training models. Also, in order to maximize coverage of the parameter space, the rates of parameters should not bear an integer multiple relationship. For repeatability in dataset generation, the phases of the triangular oscillators are reset to zero at the beginning of the dataset generation process.

4.3 File Input and Outputs

The audio file including the stimulus signal utilized as input for the audio effect can be loaded in section 'B' of the GUI. The stimulus is reproduced automatically during the dataset generation, while the GUI allows users to manually playback or loop the signal to test the audio effect while adjusting the dataset generation options. This section also lets users adjust and monitor the pre-effect gain of the stimulus signal and listen through the monitoring loudspeakers. We have chosen to use a stimulus from a file rather than an integrated signal generator as the latter may not be as comprehensive in terms of options as existing libraries or tools. Also, an integrated stimulus generator may fall short when users want to use segments of music or instrument recordings as stimuli.

When recording the output of an external hardware audio effect, the signal undergoes processing by the effect device, but it also passes through the output and input of the audio interface. While most audio interfaces provide relatively transparent sound, there could be setups where this might not hold true or applications where the contribution of the audio interface can not be overlooked. Consequently, we have added an option in section 'B' to record the stimulus file after the audio interface loopback. In this scenario, the output of the audio interface must be connected to its input. When training machine learning models of an audio effect, it is more appropriate to use the recording of the stimulus loopback as input rather than the original stimulus file.

The DGMD generates several output files. In section 'C' of the GUI, users can specify the directory where output files are saved and a prefix used for all generated files. In *Step*, *Random*, and *Manual* modes, the tool generates a separate uncompressed mono or stereo wave audio file for each parameter combination included in the dataset, recording the effect output. Audio filenames include an integer index, which serves to retrieve the associated combination of parameters from a table exported as a Comma Separated Value (CSV) file. When working with internal software plugins, the CSV file also includes the VST parameter names. Though the CSV file with parameter combinations is exported upon completion of dataset generation, users can manually export the file if the dataset generation is manually interrupted. In all modes, parameter combinations are generated internally and cannot be imported from an existing CSV file. The generator of parameter combinations for step mode is initialized when the 'Update Step Mode Combination' button in section 'A' is pressed. The generator for step mode and the memory table that holds parameter combinations for random and manual modes are cleared when the 'Reset' button is pressed. Additionally, in section 'C,' users can choose to incorporate the associated parameter values also into the names of the recorded audio files.

In *Sweep* mode, two uncompressed wave audio files are generated for each repetition of the input stimulus signal. The first file contains either a mono or stereo recording of the output processed by the effect. The second file is a multichannel wave file with many channels equivalent to the number of enabled parameters, and it includes recording at audio rare of the sweeping parameters. The filenames of both audio files include an integer index, which facilitates pairing the recorded output with the associated sweeping parameters. As previously discussed, these signals undergo a process of downsampling when the CV parameter output is not enabled. Given that parameter values change very frequently in this mode, recording them as audio signals ensures perfect alignment with the recorded output and simplifies the process of utilizing the dataset - particularly when modeling the audio effect using machine learning techniques requiring matching input-output pairs.

The generated wave audio files have a sampling rate identical to the system sampling rate, which is set in section 'D' of the GUI. The sample format is configured to a 32-bit floating point for recording the audio effect output and a 24-bit integer for parameter recording in sweep mode.

4.4 Latency Compensation

Audio effects may present some input-output latency, which varies based on whether they are analog or digital, as well as their internal signal processing architecture. In the DGMD environment, as well as in other digital audio recording environments, there are other components contributing to add significant latency.

Audio effects can present some latency between the input and output audio. This latency varies depending on whether the effects are analog or digital, and it is also influenced by the internal signal processing architecture. Within the DGMD environment, as well as other digital audio recording environments, other components contribute further, adding significant latency. For internal software plugins within Max 'Signal Vector Size' has an impact on the latency. This setting specifies the number of audio samples processed at once by Max's signal processing routines. For external hardware effect units, Max 'I/O Vector Size' accounts for additional latency, as this determines the number of audio samples buffered and exchanged with the audio interface at a time. This value usually has a minimum setting of 64 samples for most audio interfaces. Both 'Signal Vector Size' and 'I/O Vector Size' can be adjusted in section 'E' of the GUI. This additional latency introduced by the DGMD environment determines a significant misalignment between the input stimulus signal and the recorded effect output, which can be detrimental for effect modeling purposes.

The DGMD tool includes a utility to detect the latency, measured in audio samples, of the audio effect, taking into account all additional components mentioned above. When measuring, the audio effect's parameters should be set to return a signal that is as dry as possible. Measurements can be initiated in section 'A', where the detected latency is displayed. Users have the option to automatically

compensate for this latency. In this case, recordings of the effect output continue past the end of the stimulus signal by an equal number of audio samples to the detected latency. Once the recording ends, and before being written to file, a number of samples equal to the detected latency are removed from the start of the recorded audio buffer. This ensures alignment between the effect input and output signals. Users can use the latency measurement utility with different values of 'Signal Vector Size' and 'I/O Vector Size' to determine the actual latency of the effect, which should be a constant offset added to the buffering-related latency. Then, this value can be manually subtracted from the detected latency value to generate a dataset that still accounts for the internal latency of the effect.

4.5 Other Functionalities and Features

All user-defined settings, which determine the interfacing of the effect and dataset generation options, can be saved into a preset. Each preset is associated with a user-defined name and an integer index in section 'I' of the GUI. These presets can be recalled to enable reproducible dataset generation, which is essential when a single dataset generation task needs to be executed over multiple sessions. The entire collection of presets can be exported to, or imported from a file.

Audio effects can include Low-Frequency Oscillators (LFOs) that independently vary one or more parameters in the internal audio signal alteration process. In some cases, an external trigger can reset the phase of the LFOs to zero. As such, in section 'F', we have provided a utility to send to the effect a Gate signal or a MIDI note with arbitrary delay and duration each time the playback of the input stimulus begins and output recording starts. This feature allows to obtain consistent and reproducible influence of the LFOs on the dataset.

Given that the dataset generation process can be lengthy, the progress percentage is displayed in section 'A' of the GUI. The overall duration can be estimated by multiplying the number of combinations by the duration of the stimulus file. This estimation does not account for minor delays due to parameter updates and file-writing to disk. However, if an internal software plugin is being used, users can switch the audio driver to 'NonRealTime' in section 'E'. This significantly speeds up the generation process as Max's signal processing routines are executed offline.

Lastly, section 'A' of the GUI dynamically changes with the selected parameter change mode, as the names, availability, and functionality of the options may vary across modalities. Extensive information is provided to users in small pop-ups when they hover the mouse over each interactive element of the GUI.

5. CONCLUSIONS

We have introduced a tool called Dataset Generator for Musical Devices (DGMD), specifically the audio effect version. DGMD is a standalone software for generating precise and reproducible datasets from any software and hardware audio effect. This tool offers a rich set of features and functionalities which are informed by research in the field of machine learning modeling of audio effects processing. Both DGMD and its accompanying video documentation are freely available as open-source software at `https://github.com/stefanofasciani/DGMD`.

6. REFERENCES

[1] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners, "Virtual Analog Effects," in *DAFX: Digital Audio Effects*. John Wiley & Sons, Ltd, 2011.

[2] R. M. Bittner, M. Fuentes, D. Rubinstein, A. Jansson, K. Choi, and T. Kell, "mirdata: Software for reproducible usage of datasets," in *Proceedings of the 20th International Society for Music Information Retrieval (ISMIR) Conference*, 2019.

[3] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic Programming of VST Sound Synthesizers Using Deep Networks and Other Techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, 2018.

[4] G. L. Vaillant, T. Dutoit, and S. Dekeyser, "Improving Synthesizer Programming From Variational Autoencoders Latent Space," in *Proceedings of the 24th International Conference on Digital Audio Effects (DAFx)*, Sep. 2021.

[5] S. Fasciani, "TSAM: a tool for analyzing, modeling, and mapping the timbre of sound synthesizers," in *Proceedings of the 13th Sound and Music Computing Conference (SMC)*, Hamburg, Germany, 2016.

[6] L. Juvela, E.-P. Damskägg, A. Peussa, J. Mäkinen, T. Sherson, S. I. Mimilakis, K. Rauhanen, and A. Gotsopoulos, "End-to-End Amp Modeling: from Data to Controllable Guitar Amplifier Models," in *Proceedings of the 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Rhodes Island, Greece, 2023.

STOCHASTIC RESONANCE: MOLDING SOUNDS FROM NOISE

Esteban GUTIÉRREZ[1] and **Rodrigo F. CÁDIZ**[2,3]

[1] *Department of Information and Communications Technologies*, **Universitat Pompeu Fabra**, Barcelona Spain
[2] *Music Institute*, **Pontificia Universidad Católica de Chile**, Santiago Chile
[3] *Department of Electrical Engineering*, **Pontificia Universidad Católica de Chile**, Santiago Chile

ABSTRACT

This paper explores the application of stochastic resonance, a phenomenon known for enhancing signal detection through the introduction of random noise or disorder, to sound synthesis and processing. We detail the various models of stochastic resonance described in the literature and introduce a software implementation as an external for Max/MSP. Through a series of examples, including sound synthesis and processing, reducing the signal-to-noise ratio of noisy sounds, and retrieving the missing fundamental, we demonstrate the efficacy and versatility of this approach in molding sounds from noise.

1. INTRODUCTION

Stochastic resonance is a phenomenon observed in nonlinear systems where the introduction of random noise or a moderate level of disorder enhances the detection or transmission of weak or small signals. This implies that, although it may seem counter-intuitive, the addition of noise can improve the performance of a system in responding to a weak input signal.

The phenomenon of Stochastic Resonance was first introduced by Benzi, Parisi, Sutera and Vulpiani [1] in the context of a particular stochastic dynamical system, and was later formulated in many different contexts (e.g. [2], [3], [4], [5] and [6]). This phenomenon should not be understood as a single model, but rather as a technique for creating models that allows the study of different stochastic dynamical systems.

In a typical stochastic resonance scenario, a system is exposed to a weak input signal that may not trigger a response on its own. However, when a certain level of random noise is introduced into the system, it can help amplify and enhance the detection or transmission of the weak signal. This occurs because the noise can help push the system over its threshold, making it more likely to respond to the weak input signal.

Stochastic resonance is not just a theoretical phenomenon, as it has been observed in various fields, including theoretical physics [6], signal processing [7], metrology [8] and sensory biology [9]. Despite its clear potential as a signal processing tool, there are very few documented uses of these ideas in sound synthesis and processing.

In the past, efforts have been made to study and apply the Stochastic Resonance phenomenon in sound synthesis [10], [11]. However, the models proposed in these works lacked an easy-to-use interface or were impractical for real-time use due to implementation limitations.

In this paper we present the package `Stochastic Resonance`, a package for Max/MSP that implements real-time algorithms based on Stochastic Resonance techniques that we have found can be used in sound and music applications. This package was written using the Min devkit, a software development kit to develop Max/MSP externals using modern C++ code. Due to this, we were able to take advantage of the raw power of C++ and ensured that the externals in the package run as efficiently and fast as possible.

This paper also contains a theoretical discussion for all the algorithms in the package, a list of detailed sound and music potential applications and several sound examples. Thus, we believe this paper summarizes all known models and applications of the Stochastic Resonance phenomenon in the context of sound synthesis and processing.

In Section 2, we discuss all the algorithms included in the package. In section 3, we introduce the package itself together with all the implementation details. In section 4, we discuss a series of potential uses for this package in the context of sound synthesis and processing. Finally, in Section 5, we present the details of a series of sound examples that accompany this paper to demonstrate the potential of these tools.

2. MODELS

This section discusses the algorithms implemented in the `Stochastic Resonance` package. Some of them are based on our previous works [10], [11], but they underwent significant changes. Therefore, we will discuss these changes in detail. Each model is built with very different applications in mind. For a discussion on the different applications of each model and our use suggestions, see Section 4.

2.1 Threshold Stochastic Resonance

The Threshold Stochastic Resonance (TSR) algorithm, first proposed in [10], is the simplest implementation of the Stochastic Resonance phenomenon in signal processing. It

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

involves adding noise to a signal and then successively applying a constant threshold function over time. The algorithm is presented in Algorithm 1.

Algorithm 1 TSR

Input: Let x be an audio signal, $\alpha, \Theta, \sigma \in \mathbb{R}$ the attenuation, threshold and standard deviance constants, and η a noise signal with $\mathbb{E}(\eta(n)) = 0$ and $\mathbb{V}(\eta(n)) = \sigma^2$ for all time indexes n.

Output: Processed signal y.

1: Attenuate the audio signal $x_\alpha = \alpha x$.
2: Add the noise to the attenuated signal $x_\eta = x_\alpha + \eta$.
3: Apply the threshold Θ

$$y = \mathbb{1}_{|x_\eta| > \Theta} x_\eta.$$

For this algorithm to make sense in the context of stochastic resonance phenomena, it is important to choose α such that at least most of the signal x_α falls below the threshold Θ (see Figure 2), that is,

$$P(|x_\eta| > \Theta) \approx 0.$$

2.2 Multi-unit Threshold Stochastic Resonance

Due to the stochastic nature of the algorithm, applying it to a given signal multiple times will result in different outputs each time. Therefore, averaging all these outputs can provide a less noisy representation of the expected result. In the field of sensory biology, this is known as an N-unit model (see [9]). Inspired by this concept, taking the average of multiple TSR algorithm outputs will be referred to as the Multi-unit Threshold Stochastic Resonance (MTSR) algorithm. A version of this algorithm with a different name was also proposed in [10]. The algorithm is presented in Algorithm 2.

Algorithm 2 MTSR

Input: Let x be an audio signal and $N \in \mathbb{N}$.

Output: Processed signal y.

1: Make N copies of the original signal, that is, $x_k = x$, for $k = 0, \ldots, N-1$.
2: Apply the TSR algorithm to each copy using the same parameters $y_k = \text{TSR}(x_k)$.
3: Compute the average of all the new signals

$$y = \frac{1}{N} \sum_{k=0}^{N-1} y_i.$$

2.3 Spectral Threshold Stochastic Resonance

Applying noise and a threshold function to the spectrum of a signal requires careful consideration. On one hand, the perception of loudness is strongly dependent on frequency (see e.g., [12], [13]), so the addition of noise must be done with attention to this fact. On the other hand, applying a threshold function requires an ordering notion that is not inherent in the complex numbers, which is the natural space for the spectrum. An easy solution for this problem would be applying the threshold to the real or imaginary part of the spectrum, but this lacks perceptual interpretation. Another way of solving this would be applying the threshold function to either the magnitude spectrum or the phase spectrum. Since our goal is to implement these algorithms in real-time, we encounter the challenge of dealing with spectrum frames. Manipulating the phase while noise is involved can lead to crackling sounds due to the inconsistency of the phase across frames. Because of the latter, the only candidate left is the magnitude spectrum, however there is still a problem: it is built-in in the nature of the Fourier Transform that the magnitude spectrum of a periodic signal will show at least some decay. This outcome is a consequence of the Riemann-Lebesgue Theorem, suggesting that a constant threshold function may exhibit a preference for the lower part of the spectrum in certain sound contexts. Additionally, the threshold function must depend on the input signal's nature. In order to solve these issues, the Spectral Threshold Stochastic Resonance (spectral TSR) algorithm adds noise balanced according to an equal-loudness contour, and replaces the threshold constant for a function that can be built by the user (see subsection 3.2 for implementation details). The algorithm itself is presented in Algorithm 3.

It is important to note that this algorithm is very different to the one presented in [11]. In such work, the Spectral Stochastic Resonance algorithm presented generates completely random resonances. More precisely, given a set of parameters, the SSR algorithm applies plain noise to the magnitude spectrum and then independently applies a random threshold to each bin. As a result, it does not account for the equal-loudness effect of the added noise or the characteristics of the sound when determining the threshold.

2.4 Spectral Multi-unit Threshold Stochastic Resonance

Similar to the time domain, the Spectral TSR algorithm can be applied multiple times and the average can be computed to reduce the noisy nature of the expected output. The algorithm itself is presented in Algorithm 4.

3. IMPLEMENTATION IN MAX

To implement our algorithm efficiently within a friendly environment, we opted to utilize the Min dev-kit, a software development kit containing an example package with the current best practices for package creation using modern C++ code. Specifically, with the Min dev-kit, we successfully built a comprehensive package for Max/MSP that is very user-friendly and can be freely downloaded from our GitHub repository [1] .

The resulting package was named `Stochastic Resonance`, and it encompasses all the algorithms presented in Section 2, implemented as independent externals.

[1] The GitHub repository can be accessed here: `https://github.com/cordutie/Stochastic-Resonance.git`

Algorithm 3 Spectral TSR

Input: Let x be an audio signal, $\alpha, \sigma \in \mathbb{R}$ the attenuation and standard deviance constants, Θ the threshold function, S the sampling rate, $\{\eta_{j,k}\}_{j,k\in\mathbb{N}}$ a sequence of random variables with $\mathbb{E}(\eta_{j,k}(n)) = 0$ and $\mathbb{V}(\eta_{j,k}(n)) = \sigma^2$ for all j, k, n, ω a window function and m, h the frame and hop sizes respectively.

Output: Processed signal y.

1: Attenuate the audio signal $x_\alpha = \alpha x$.

2: Define each frame of the signal by the sequence x_k given by

$$x_k(j) = x_\alpha(j + kh)\omega(j), \quad j = 0, \ldots, m-1.$$

3: Compute the magnitude and phase spectrum of the frame using the DFT

$$\mathcal{F}(x_k)(j) = M_k(j)\exp(iP_k(j)).$$

4: **for** each frame k and $j \in \{0, \ldots, m-1\}$ **do**

5: Compute the frequency of the bin $f_j = j \cdot \dfrac{S}{m}$.

6: Correct the loudness of the noise according to the frequency of the bin using a contour curve C

$$\tilde{\eta}_{j,k} = \eta_{j,k}C(f_j).$$

7: Add the corrected noise to the magnitude spectrum

$$M_{k,\eta}(j) = M_k(j) + \tilde{\eta}_{j,k}.$$

8: Apply the threshold function

$$\tilde{M}_k(n) = \mathbb{1}_{M_{k,\eta}(n)>\Theta(f_n)}M_{k,\eta}(n).$$

9: **end for**

10: Build new frames using the IDFT and the new magnitude spectra and the same phase spectra

$$y_k = \mathcal{F}^{-1}(\hat{M}_k \exp(iP_k).$$

11: Overlap and add the new frames y_k to obtain a new signal y.

Algorithm 4 Spectral MTSR

Input: Let x be an audio signal and $N \in \mathbb{N}$.

Output: Processed signal y.

1: Make N copies of the original signal, that is, $x_k = x$, for $k = 0, \ldots, N-1$.

2: Apply the Spectral TSR algorithm to each copy using the same parameters $y_k = \text{Spectral TSR}(x_k)$.

3: Compute the average of all the new signals

$$y = \frac{1}{N}\sum_{k=0}^{N-1} y_i.$$

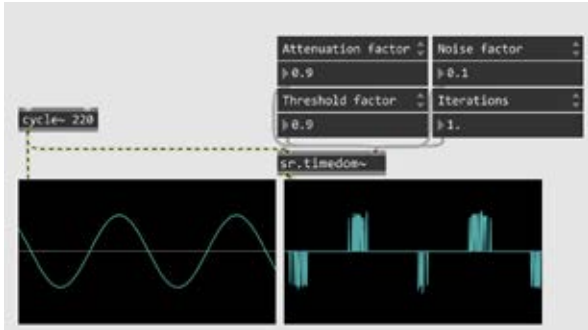


Figure 1. `sr.timedom` object running on Max. The parameters of the algorithm are passed to the object using the `attrui` object for clarity.

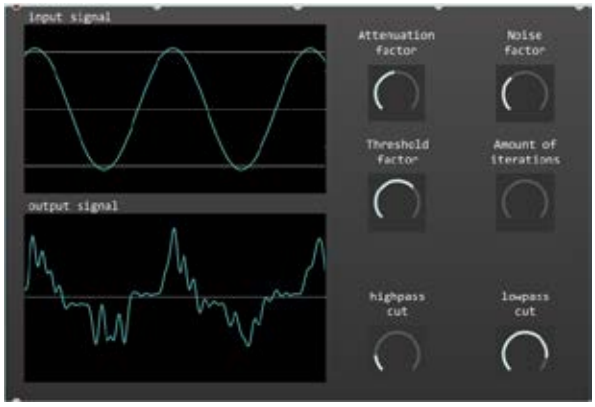


Figure 2. Example patch using the `sr.timedom~` object loaded with the `bpatcher` object. All the parameters are controllable using knobs. The threshold and attenuation are visualized in the input signal and a pair of knobs that control the filters were placed in the bottom right of the patch.

A concise explanation of the usage for each object within the package is provided here.

3.1 `sr.timedom`: Stochastic Resonance in the time domain

Since the TSR algorithm can be thought of as applying the MTSR algorithm with $N = 1$, we decided to implement both in only one object called `sr.timedom`. The left inlet is designated for the input signal, while the right one receives parameters as a list of numbers (see Figure 1). Additionally, the object has a built-in method to print the parameters used to the console when it receives a bang.

For an easy-to-use experience, we highly recommend opening the help of the object. Inside, we've included a subpatch read through `bpatcher` to provide a simple but intuitive GUI that implements a visualization of the attenuation factor, threshold function and both a low pass and a high pass filter to reduce noise (see Figure 2).

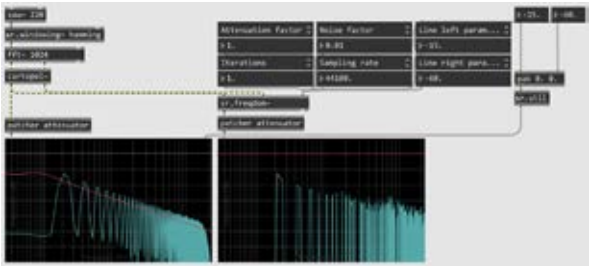


Figure 3. `sr.freqdom~` object running on Max. The parameters of the algorithm are passed to the object using the `attrui` object for clarity. The auxiliary objects `sr.windowing~` and `sr.util` are used to apply windows and compute the threshold function for plotting.

3.2 `sr.freqdom`: Stochastic Resonance in the frequency domain

Again, both algorithms Spectral TSR and Spectral MTSR are implemented in the same object named `sr.freqdom`. Since any implementation of the overlap-and-add algorithm requires precise use of a buffer and we wanted to make this package as easy to use as possible, we decided to utilize the built-in Max/MSP environment created by the `pfft~` object. This object is capable of applying a given window, computing the both the FFT and IFFT, and applying the overlap and add algorithm. Thus, the only part left to correctly implement our algorithms is to add the corrected noise and apply the threshold function.

To correct the noise, we applied a contour curve using the C-weights [14], known for its effectiveness in measuring loudness for noisy sounds. For the threshold function, we decided to implement a method in the `sr.freqdom~` object that builds a threshold function that behaves as a line in the range from 100Hz to the Nyquist frequency when viewed as a function from the log-frequency domain to amplitude measured in decibels, and that behaves as a constant for frequencies under 100Hz (see Figure 4). This convention allows the user to rapidly create threshold functions that suit the input signal and also add a component that can be used creatively.

The object itself has two signal inlets and outlets for magnitude and phase spectra, and a third inlet for receiving parameters (see Figure 3). Additionally, it has a built-in method to print the parameters used to the console when it receives a bang.

Like its time-domain counterpart, we included a subpatch in the repository. When read through `bpatcher`, it provides a simple yet intuitive GUI that implements a visualization of the attenuation factor and threshold function together with the magnitude spectrum of both the input and output (see Figure 4).

4. SOUND AND MUSIC APPLICATIONS

The `sr.timedom~` and `sr.freqdom~` objects are versatile tools capable of processing various signal types, making them suitable for both creative and analytic endeavors. In this section, we will briefly discuss some in-

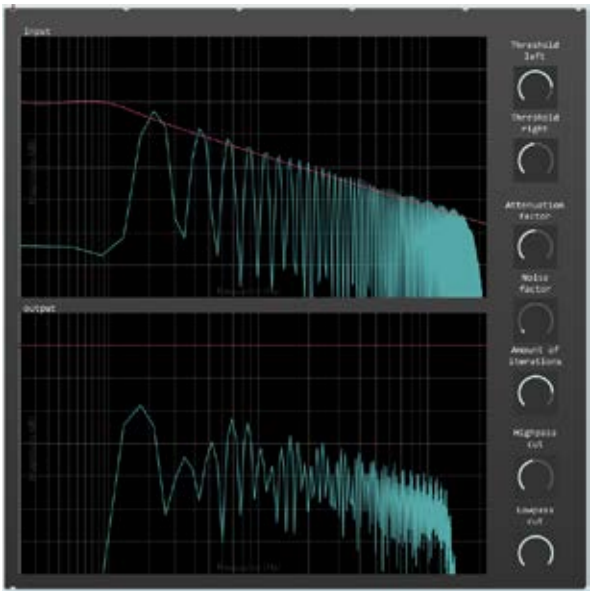


Figure 4. Example patch using the `sr.freqdom~` object loaded with the `bpatcher` object. All the parameters are controllable using knobs. The threshold and attenuation are visualized in the magnitude spectrum of the input signal and a couple of filters are implemented in the bottom right of the patch.

triguing ways of employing these objects. For specific examples, refer to Section 5.

4.1 Sound Synthesis

While the `sr.timedom~` and `sr.freqdom~` objects function as signal processors, when a simple signal such as a sine, saw, square, triangle, pulse wave, or white noise is used as input, we conceptualize the result as a novel form of sound synthesis. In such case, one must consider the type of input signal as an extra parameter together with all the parameters of the algorithms implemented. A vast array of sound types can be generated using this method. For some examples, see Section 5.1

4.2 Recovering Weak Signals

Recovering weak signals represents a prominent application of the Stochastic Resonance phenomenon in signal processing. To be more specific, empirical evidence has demonstrated that introducing noise alongside the application of a detection function can result in the identification of signals that are typically undetectable (see [8]). However, this phenomenon represents more of an analog application of Stochastic Resonance, given that issues related to the detection and measurement of signals are not a problem in the digital domain. Consequently, this application falls outside the scope of our implementation. Nevertheless, empirical validation has shown that this procedure can enhance the signal-to-noise ratio under certain circumstances (see [7]). This capability finds particular relevance in music applications, making our package suitable for such purposes. It is important to note, however, that the results

achievable through these techniques in this context significantly lag behind the outcomes attainable with contemporary denoising and source separation methods. For specific examples, refer to sub-section 5.2.

4.3 Missing Fundamental Retrieval/Ghost Stochastic Resonance

The missing fundamental is a well-known psychoacoustic phenomenon where the listener can perceive the pitch of a harmonic signal with an absent first harmonic. Given its psychoacoustic nature, grasping this phenomenon involves understanding how our brain processes information transmitted by our auditory system. In this context, Stochastic Resonance models have been proposed to understand our pitch perception in certain scenarios. Specifically, empirical evidence has shown that for a signal made of a harmonic complex of pure tones with a missing fundamental, the application of Stochastic Resonance techniques can lead to the sudden emergence of the fundamental [15]. Given that the Stochastic Resonance phenomenon is used to model the emergence of a pitch that was not originally in the signal, this effect has also been termed Ghost Stochastic Resonance.

The Ghost Stochastic Resonance effect can be easily utilized to create new pitches that are consonant with the input signal, as in the case of the missing fundamental described before. Due to this, we believe it has immense creative potential, and our package can be successfully used to implement it. For some examples, see sub-section 5.3.

5. EXAMPLES

Alongside this article, we have created some videos [2], demonstrating a series of sound examples built using the `Stochastic Resonance` package. A brief explanation for each of the sound examples can be found in this section.

5.1 Sound Synthesis and Sound Processing

For the examples in this subsection, we created two Max for Live devices. One of them is a MIDI synthesizer that wraps a sound wave generator processed through the `sr.timedom~` object and an amplitude envelope module (see Figure 5). The other one is simply a wrapper for the `sr.freqdom~` object (see Figure 6). Both Max for Live devices can be found in our repository.

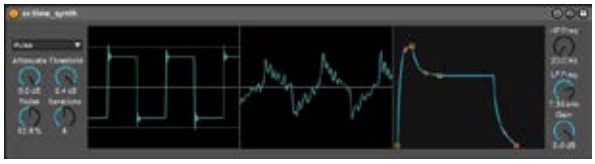


Figure 5. Max for Live device using the `sr.timedom~` object as a MIDI synthesizer.

[2] All videos are available in the GitHub repository `https://github.com/cordutie/Stochastic-Resonance`

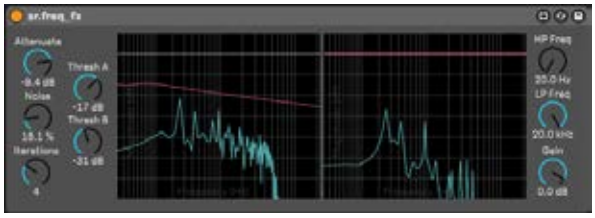


Figure 6. Max for Live device using the `sr.freqdom~` object as a sound effect.

5.1.1 *Stochastic Resonance-Based Bass Synthesizer*

In this example, a bass sound is synthesized using a sawtooth sound wave processed with the `sr.timedom~` object. Initially, a simple sawtooth can be heard, followed by aggressive processing with our object. Then, the `sr.freqdom~` is used to add distortion. Finally, the synthesized sound is contextualized with a drum set and an electric piano.

5.1.2 *Pushing Dynamics and Resonances of an Electric Piano*

A sound effect customary for electric pianos since the 1970s is the tremolo. In this example, an electric piano sample with a tremolo is processed using the Max for Live wrapper for the `sr.freqdom~` object. The sample is first played, then processed with our object using a threshold function with decay. This process pushes the dynamics of the tremolo effect with a gate-type model in the frequency domain. Noise is then added in the frequency domain, introducing stochastic resonances in the higher harmonics of the sound. Finally, all elements are combined with a drum set and a electric bass guitar.

5.2 Recovering Weak Signals

5.2.1 *Sawtooth + Noise*

In this example, a signal created by summing a sawtooth wave with a frequency of 220Hz and amplitude -12dB, and white noise with an amplitude of 0dB, is processed using the `sr.timedom~` object and a specific set of parameters to increase the signal-to-noise ratio. Additionally, a low-pass filter is applied to both the original and processed signals for a fair comparison. The result shows that the processed signal has a noticeably lower noise floor.

5.2.2 *Street Recording*

In this example, a reasonably noisy street recording is processed using a compressor together with the `sr.timedom~` object and a specific set of parameters to increase the signal-to-noise ratio. Additionally, a low-pass filter is applied to both the original and processed signals for a fair comparison. The results show that the processed signal has a distinct noise floor. Specifically, the gain of the noise floor seems to have decreased, but several crackling noises have appeared as a trade-off. As mentioned in Section 4, the results obtained using this method significantly lag behind the outcomes attainable with contemporary denoising and source separation methods.

5.3 Missing Fundamental Retrieval/Ghost Stochastic Resonance

5.3.1 Recovering missing fundamental for a complex of sinusoids

In this example, a complex of 8 sinusoids is processed in real-time using the `sr.timedom~` object. The frequencies of the complex of sinusoids follow a harmonic distribution with a missing fundamental, i.e., $2f, 3f, \ldots, 9f$ where f is the fundamental frequency. Additionally, the amplitudes and phases of each sinusoid in the complex are manipulated in real-time. The results show that the missing fundamental can be retrieved with ease, which is consistent with the simulations run in [15].

5.3.2 Building missing fundamental for a complex of voices

For this example, we used a set of previously recorded samples containing a single note sang by a female singer [3]. These samples are played at different rates so that their fundamentals follow a harmonic distribution with a missing fundamental, as in 5.3.1. The results show that applying the `sr.timedom~` object introduces a fundamental with ease in every scenario. Moreover, since the original signal consists of a sum of signals that resemble the harmonics of a sound but already have harmonics, some other resonances appear in the spectrum corresponding to what seems to be the fundamentals of the harmonics of these complex sounds. When a low-pass filter is applied to the resulting signal, a new synthesized sound that resembles some characteristics of the original voice appears.

6. CONCLUSIONS

In this paper, we propose a series of algorithms based on the Stochastic Resonance phenomenon for use in sound, accompanied by a theoretical discussion and exploration of their potential applications in sound synthesis and processing.

Our contribution goes beyond mere theory, providing a user-friendly and efficient environment for musicians, sound designers, and researchers. The `Stochastic Resonance` package, implemented in real-time in both Max/MSP and Live as a Max for Live device, streamlines the utilization of our proposed algorithms, seamlessly integrating them into any workflow. The package can be freely downloaded from our GitHub repository. Additionally, we present a series of examples to showcase the practical applications of the Stochastic Resonance package in sound synthesis and processing.

In summary, by amalgamating theoretical insights with practical examples, this paper serves as a comprehensive resource for those keen on unlocking the creative potential of the Stochastic Resonance phenomenon in the realm of sound and music.

Acknowledgments

This research was partially funded by ANID Fondecyt Grant #1230926, Government of Chile.

[3] The samples were provided personally by their author, Jessica Torrey.

7. REFERENCES

[1] R. Benzi, A. Sutera, and A. Vulpiani, "The mechanism of stochastic resonance," *Journal of Physics A: Mathematical and General*, vol. 14, p. L453, 01 1981.

[2] L. Gammaitoni, "Stochastic resonance and the dithering effect in threshold physical systems," *Phys. Rev. E*, vol. 52, pp. 4691–4698, Nov 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.52.4691

[3] ——, "Stochastic resonance in multi-threshold systems," *Physics Letters A*, vol. 208, no. 4, pp. 315–322, 1995. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0375960195007316

[4] L. Gammaitoni, F. Marchesoni, E. Menichella-Saetta, and S. Santucci, "Stochastic resonance in the strong-forcing limit," *Phys. Rev. E*, vol. 51, pp. R3799–R3802, May 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.51.R3799

[5] L. Gammaitoni, F. Marchesoni, and S. Santucci, "Stochastic resonance as a bona fide resonance," *Phys. Rev. Lett.*, vol. 74, pp. 1052–1055, Feb 1995. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevLett.74.1052

[6] L. Gammaitoni, P. Hänggi, P. Jung, and F. Marchesoni, "Stochastic resonance," *Rev. Mod. Phys.*, vol. 70, pp. 223–287, Jan 1998. [Online]. Available: https://link.aps.org/doi/10.1103/RevModPhys.70.223

[7] M. Ueda, "Improvement of signal-to-noise ratio by stochastic resonance in sigmoid function threshold systems, demonstrated using a cmos inverter," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 10, pp. 1978–1985, 2010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0378437110000890

[8] A. Palonpon, J. Amistoso, J. Holdsworth, W. Garcia, and C. Saloma, "Measurement of weak transmittances by stochastic resonance," *Opt. Lett.*, vol. 23, no. 18, pp. 1480–1482, Sep 1998. [Online]. Available: https://opg.optica.org/ol/abstract.cfm?URI=ol-23-18-1480

[9] J. J. Collins, C. C. Chow, and T. T. Imhoff, "Stochastic resonance without tuning," *Nature*, vol. 376, no. 6537, pp. 236–238, Jul 1995. [Online]. Available: https://doi.org/10.1038/376236a0

[10] R. Cadiz and P. De la Cuadra, "Stochastic resonance sound synthesis," *International Computer Music Conference, ICMC 2008*, 08 2008.

[11] R. Cadiz and P. de la Cuadra, "Spectral stochastic resonance sound synthesis," *International Computer Music Conference, ICMC 2010*, pp. 353–356, 01 2010.

[12] D. W. Robinson and R. S. Dadson, "A re-determination of the equal-loudness relations for pure tones," *British Journal of Applied Physics*, vol. 7, no. 5, p. 166, 1956.

[13] Y. Suzuki and H. Takeshima, "Equal-loudness-level contours for pure tones," *The Journal of the Acoustical Society of America*, vol. 116, no. 2, pp. 918–933, 08 2004. [Online]. Available: https://doi.org/10.1121/1.1763601

[14] International Electrotechnical Commission (IEC), "Electroacoustics - Sound level meters - Part 1: Specifications," IEC, Geneva, Switzerland, Tech. Rep. IEC 61672-1, 2002.

[15] P. Balenzuela, H. Braun, and D. Chialvo, "The ghost of stochastic resonance: An introductory review," *Contemporary Physics - CONTEMP PHYS*, vol. 53, 10 2011.

SPECULATIVE MACHINE LEARNING IN SOUND SYNTHESIS

Luc DÖBEREINER (doebereiner@iem.at) and **David PIRRÒ** (pirro@iem.at)

Institute of Electronic Music and Acoustics, **University of Music and Performative Arts Graz**, Austria

ABSTRACT

This paper explores the intersection of technical, conceptual, and artistic aspects of an ongoing engagement with machine learning (ML), specifically with generative ML in computer music, emphasizing the dynamics of the learning process over its outcomes. Unlike most approaches that prioritize generating "realistic" instrumental or vocal sounds from large datasets, this work integrates the sound synthesis model directly into the learning system, allowing neural networks to produce sound in real-time by continuously adapting and predicting. This method treats the learning process's behavior as a generative sound synthesis process, making it perceptible. The research delves into how various parameters, such as learning rate and ridge regularization, influence the generation of sonorous behavior, and speculates on the possibilities of a radical reformulation of standard learning functions with a particular interest in complex, chaotic, and relational properties of adaptive computational processes. The paper outlines the motivation behind this exploration, introduces the "Speculative Sound Synthesis" project as its context, presents two case studies to illustrate the approach, describes several experimental artistic applications, and concludes with reflections on the findings and future research directions.

1. INTRODUCTION

Over the past decade, AI and in particular machine learning techniques and algorithms have gained a central role in computer music and electronic music composition. Much of the attention in machine learning in sound synthesis has been focused on clearly defined processing tasks such as source separation [1], timbre transfer [2], singing voice synthesis [3], or parameter generation for synthesis using known methods, such as frequency modulation [4, 5]. While some approaches have utilized latent space representations [6, 7] and recurrent neural network dynamics in conjunction with granular synthesis and echo state networks [8] to explore novel approaches, the focus of most research in this area has largely been on the simulative generation of "realistic" vocal or instrumental sounds.

As Jonathan Impett writes, "An AI is essentially a memory machine" [9, p. 225]. One way to understand deep learning systems is as a form of memory of certain past givens. It then becomes an artistic question of how to inscribe, recollect, overwrite and navigate within such a compressed memory. Current synthesis models leveraging deep learning mostly utilize extensive datasets for training and, crucially, they separate the training phase from the inference or prediction phase. The memory is built and later activated. The training phase is therefore regarded as a preparatory step that is carried out with a view to optimal generalization for another task. However, the training phase can also be seen as an adaptive computational process, which itself harbors musically and sonically interesting dynamics and potentials. The two approaches described in this paper are technically and conceptually very different, but both take the adaptive dynamic of machine learning as their starting point. The approach described first speculatively challenges common autoregressive technologies of time-series prediction, while the second approach embraces the emergent behavior or recurrent networks of interacting and independently learning neurons. In both approaches, the learning process is not used for the sake of some external optimization goal, but is itself part of the synthesis process. In other words, we are interested in the specific forms of continuous memory inscription and reactivation as a form of computational performance.

1.1 Speculative Sound Synthesis

The systems and approaches described in this paper were developed as part of the research project Speculative Sound Synthesis, [1] which aims to reconsider the relationship between technology in computer music and artistic practice. It seeks to uncover the aesthetic possibilities of sound synthesis that may go unnoticed if technology is viewed solely as a functional tool for control and execution. In contrast, the approaches described in this text aim to experimentally and exploratively open up machine learning technologies, speculate on their paradigms, and expose their computational materiality for aesthetic experience, musical performance, interaction, and composition.

The models described in this text are *speculative sound models*, which is a central methodological concept of this project. They materially manifest speculations with technological constructs that challenge instrumentality, control, and representation. This is why hyperparameters, such as network structure, learning rate, ridge parameters, and activation functions, become central artistic material instead of being merely subserviant functional components for the realization of external and predefined tasks.

[1] https://speculative.iem.at

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

1.2 Speculative Propositions

With these aesthetic, technological and methodological considerations in mind, we formulate a set of basic speculative propositions that inform the two approaches described in the remainder of this paper:

1. The adaptive training stage is an essential element of the audible dynamics (online learning) and does not exclusively occur before sound synthesis. Training, learning, and inference all happen during performance. The aim is to create a machine learning model that is perceivable and tangible in both sonic and performative terms. It is a memory in touch with its input and output, learning while it is performing.

2. The sound synthesis model is an integral part of the machine learning system, with the neural network directly producing time-domain signals in "real-time". The synthesis method is thus not external to the model, it is not controlled by the model but what is audible is itself and output of the model.

3. In line with the previous presuppositions, the input data is intentionally kept small. The usage of big data is entangled with structures of power and economic exploitation, with interpolating reproduction of memorized data points. In contrast we regard the small data approach as a form of artistic reappropriation of technology and as a method to expose its learning dynamics rather than to generalize as well as possible with regard to input data. Small datasets can be influenced by the biases and tendencies of musicians and composers, which can be part of the compositional process, and the agency of training datasets can thus become more tangible.

2. AUTOREGNN: AUTOREGRESSIVE FEED-FORWARD NEURAL SYNTHESIS

AutoRegNN [2] is an autoregressive feed-forward neural network that predicts the value of the next sample based on previously generated samples. In principle this method is similar to the seminal WaveNet [10] architecture, although it works with much smaller amounts of input data, does not use convolutional layers and is intended for online training while generating sound. It is implemented as a unit generator (UGen) for *SuperCollider*. It consists of an input layer of variable size, a hidden layer also of variable size, and an output layer, which is always of size one. At each data point, the network utilizes a sequence of previous data points and predicts a single output sample. Autoregression involves using the network's own past predictions as inputs to predict subsequent time steps, thus creating a feedback loop. For each audio sample AutoRegNN performs two separate forward passes, one using external input data, i.e. an input signal whose dynamics are to be learned, and as second pass using its own past output data, the output of which is used as the direct audible output of the system. The output of the first pass, using the external input, is used to compute the loss and thus to adjust the weights and biases using backpropagation. In addition to the squared error the loss function also contains a ridge regression regularization term proportional to the square of the magnitude of the coefficients. This form of regularization penalizes large weights and thus draws them closer to zero. This stabilizes the learning phase and serves to prevent exploding gradients, which is especially important if the learning rate is modulated and may intentionally be used to generate chaotic behavior. The loss function is:

$$L = (y - \hat{y})^2 + \lambda \sum_{j=1}^{M} w_j^2$$

Where:

- L is the loss function, combining squared error with an L2 regularization term.
- y is the actual value for the single sample (external input).
- $\hat{y}$ is the predicted value by the neural network for the single sample.
- λ is the regularization parameter, controlling the strength of the weight penalty.
- M represents the total number of weights in the neural network.
- w_j denotes the j-th weight in the neural network.

The weights are adjusted after the production of each single sample in order to minimize the loss using backpropagation. At initialization the size of the input layer and the size of the hidden layer (number of perceptrons) can be set. There are two activation functions used throughout the network: "tanh" and "sine" (for details, see subsection 2.1.2). The size of the input layer determines the number of previous samples considered. This also conditions the temporal relationships and frequencies that the network learns to model. In place of the convolutional layers in WaveNet, we have tried various methods of downsampling and filtering the input in order to allow for temporal relations and lower frequencies to be learned, while keeping the input data small. The present iteration employs a cascade of one-pole low pass filters that process the input before it is fed into the input layer. The coefficients for the filters are calculated using the formula $e^{-1/(2^k)}$, where k ranges from 1 to n (n represents the input size). Each succeeding filter is fed into the following one which has proven to increase sensitivity across a wide range of frequencies while reducing the input size.

The model depends strongly on its random initialization, especially if λ, the regularization paramter, is small. A significant parameter to consider is the learning rate. High learning rates can result in chaotic dynamics and rapidly changing timbral output, whereas very small learning rates can generate very gradual changes in pitch and timbral development. The learning rate is thus not treated as hyperparameter to be optimized with regard to the model's ability

[2] The code can be downloaded here: `https://git.iem.at/lucdoebereiner/autoregnn`

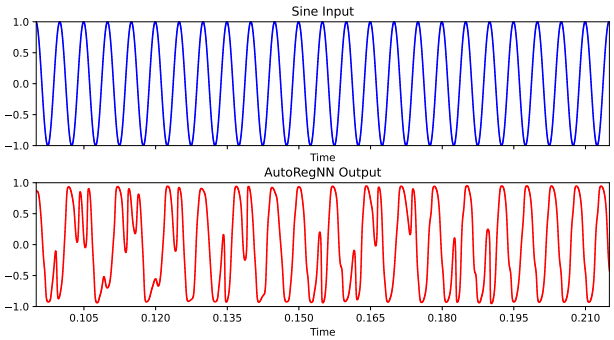


Figure 1. Learning a 200 Hz sine wave signal.

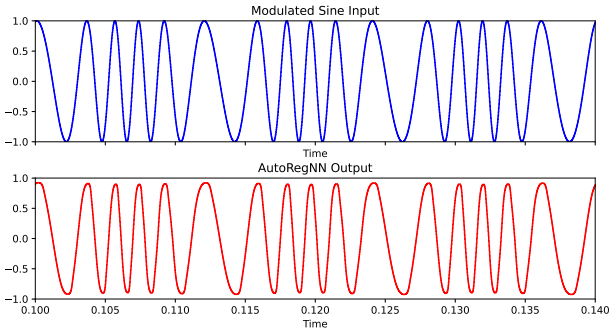


Figure 2. Learning a frequency modulated sine wave.

to generalize, but much rather as a parameter that dyanmically influences the musical and timbral behavior of the synthesis process. Hence, the learning process merges with the actual sound synthesis, and its artifacts and materiality, including chaotic and periodic behavior, serve as both artistic and musical material.

2.1 Examples and Artistic Experiments

This section presents several examples and artistic experiments that illustrate the capabilities of AutoRegNN. The code and audio and video recordings for each of these are available online. [3]

2.1.1 Examples

Fig. 1 shows 120 ms of the input and output of a sine wave with a frequency of 200 Hz fed input an AutoRegNN with and input size of 12, a hidden layer of size 8, a learning rate of 0.1 and a ridge parameter of 0.0001. It demonstrates the initial adaptation as the output is initial somewhat unstable but progressively takes on the input frequency. Lower learning rates lead to longer phases of initial adaptation.

Fig. 2 shows 40 ms of a frequency modulated sine wave. The main difference with regards to the example show in fig. 2 is that the learning rate is much higher (0.6) and the ridge parameter is also much higher (0.01). This leads to a very quick and stable adaptation.

Fig. 3 shows 20 ms of using a 180 Hz sawtooth wave as input. In this example, the learning rate is also modulated using a 180 Hz sawtooth wave with values between 0.01

[3] `https://speculative.iem.at/docs/speculations/Experiments_thoughts/autoregnn/`

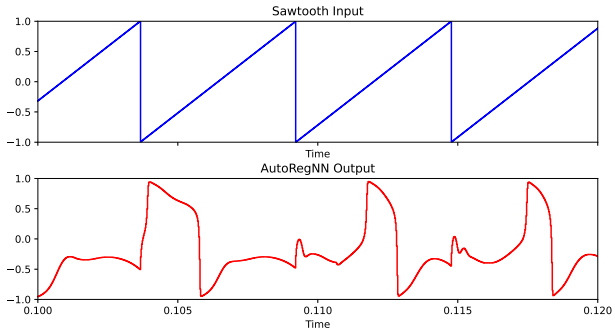


Figure 3. Learning a 180 Hz sawtooth wave with modulated learning rate.

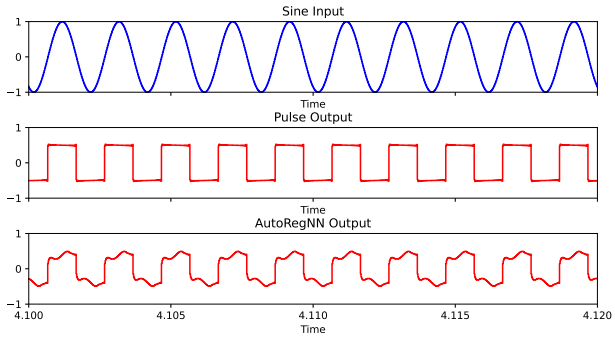


Figure 4. Interpolating between two sets of learned network parameters.

and 1. The ridge parameter is relatively small at 0.00001 in order to allow for a higher degree of unstable and chaotic behavior. Modulating the learning rate can create interesting rich timbres that are still coupled to the input, while exhibiting a certain independence.

There are a number of slightly different variants of AutoRegNN, one of which takes two audio inputs and learns two sets of network parameters (weights and biases), while using a linear interpolation between both sets of parameters for the generation of the output. Fig. 4 shows 20 ms of such an interpolation between sets trained on a sine wave and a pulse wave, both with a frequency of 500 Hz. In this example, the output is generated with the mean values of the trained weights and biases. The learning rate is 0.5 and the ridge parameter is 0.001.

2.1.2 Artistic Experiments

The artistic experiments sketched in this sections serve to document discoveries made during the conceptual and technological deveopment of AutoRegNN. However, the are to be understood as speculative starting points. They point towards possible experimental artistic scenarios and warrant further exploration, expecially with regard to their unique performative affordances and musical potential.

There are several variants of the AutoRegNN SuperCollider ugen that read and write their model parameters form and into external buffers. This allows an interaction with the weights and biases during training and prediction. Any external process that manipulates buffers can be used to

interfere, couple or process the learned paramemeters. It thus further exposes the inner workings of the neural network and can be used to turn it itself into an instrument for sound performance. In one artistic experiment, we have created a multi-slider graphical user interface that exposes all weights and biases as indiviual faders. The direct manipulation of the internal parameters allows for speculative exploration of the network. One can find thresholds at which the system jumps from one type of behavior to another. In conjunction with the control of the learning rate and the ridge regression, one can thus explore the possibilities of the neural network by changing the individual weights. One can find neighboring states while negotiating between manual manipulation and the counterforce of learning (controlled by the learning rate). This creates a kind of speculative co-agency between machine learning and manual exploration.

The activation function variation is a productive feature for sound and behavior. AutoRegNN currently allows interpolation between two activation functions: tanh and sine. Tanh produces more predictable and stable results, while sine is used similarly to a sine map and exhibits chaotic dynamics, but remains derivable and compatible with gradient descent and backpropagation for optimization. It is possible to interpolate between the two activation functions independently for each neuron of the network. This allows for the production of adaptive sounds and sound dynamics that are both chaotic and highly complex in terms of timbre.

Another promising application that produces complex, chaotic, emergent sound textures with intriguing gesturality can be produced by constructing feedback networks of multiple AutoRegNN units. For example, we constructed circular networks of four AutoRegNN units, one of which is initialized with and external signal (noise, e.g.) at the beginning. The initial signal fades out and the generators continue to excite and imitate each other. With different ways of mixing, filtering and delaying the feedback signal between the four generators, a wide variety of distinctive sonic and musical dynamics are created.

We also extended the idea of feedback networks of AutoRegNN generators and conducted experiments with models that incorporate acoustic feedback, where microphone signals provide the input training signal. We also conducted experiments with different interconnected models that interact through microphones in space and by attaching transducers and contact microphones to percussion instruments, especially drums and cymbals. Very small learning rates have been shown to produce interesting gradual changes in pitch and timbre. The ability to mimic their input depends on the type of physical instrument to which each generator is connected via playback through a transducer. For example, when certain resonant frequencies are reached, the gradually evolving timbre can suddenly fall into unstable transients and destabilize the whole network of generators. Moreover, the physical setup enables interactions between materials where the algorithmic, the object and a musician's agency are intertwined, such as through the damping, striking, and wiping of a drum.

Moreover, AutoRegNN has recently been employed by the composer Farzaneh Nouri in an octophonic interactive sound installation titled. *Embedded/Embodied*[4] In conjunction with several layers of AI sound recognition, generation, and communication, the work uses multiple instances of AutoRegNN as a multi-agent system fed by microphone inputs located in different parts of the installation's environment. The agents capture and process sounds from their environment and are used to generate emergent responses that shape and reveal their understanding. Nouri describes the work as an acoustemological exploration.

3. FORGET LEARNING

Most machine learning systems based on neural networks share a few similar paradigms that vary depending on the specific implementation. The experimental system we describe below is the result of a speculative approach that questions some of these paradigms and seeks alternative approaches to the use of neural networks. In its implementation, we followed both scientific and aesthetic criteria.

It is crucial to understand that this case study does not aim to produce a novel system that successfully performs a set of predefined tasks. The system we describe here does not successfully learn patterns or produce categorizations of some input signals. The neural network we develop here is studied in terms of a sound generator that perceptually reveals a specific temporality of the learning process. The dynamics arise from the evolution of the whole system, seen as a collection of individual neurons, each following the same set of rules.

The starting point for this speculative system has been the exploration of the connection between neural networks, as they are interpreted and implemented in machine learning applications, and the computational modeling of their physiological counterparts, biological neural networks, whose organization and structure serve as inspiration for the former. Specifically, from the similarities and, especially, the differences in the understanding of the concepts of learning and adaptation, ideas and inspirations are derived that are at the center of this *speculative sound model*. These inspirations are further put to the test and refined in the practical implementation.

Learning, in the context of neural dynamics is understood as a process of *adaptation* [11, 12] to stimuli that each neuron in a network undergoes. Neurons learn by adapting their activity pattern in response to the input stimuli they receive: in particular, neurons adapt by *reducing* their response to repeated stimuli. The well-known auditory masking effects can be seen as an example of this adaptation mechanism: contrary to some understandings of neural network learning, this means that repeated stimuli are masked or "filtered" by the neurons. Thus, when a neuron (or a network) recognizes a learned stimulus, it does not as one might expect, indicate this with a peak of activity or a strong response. That is, "any adaptive system converges to a state in which any kind of stimulation ceases" [13].

[4] https://2024.sonicacts.com/programme/embedded-embodied

Thus, from the perspective of neural dynamics, for one neuron learns or adapts by finding a configuration of its internal behavior that is less sensitive to the recurrent stimuli and thus more sensitive, i.e. producing a stronger response to input signals that deviate from known patterns.

This mechanism is further developed in recent advances in neuroscience, particularly in the context of the theory of predictive coding. This theory sees the brain as essentially a predictive machine, constantly generating "mental models" of the environment: these models are used to infer or predict input sensory signals, and are then compared with the actual signals received to construct a new model. It has been proposed to apply this perspective not only to the brain as a whole, but also to individual neurons. Thus, *learning* for a single neuron means adapting the weights it applies to its inputs in order to be able to *predict* future inputs. Predicting in this context means minimizing the "surprise" defined as the difference between the actual and the predicted or adapted inputs [14]. This learning rule theory seems to be supported by neural metabolic mechanisms and in particular by the so-called "lazy neuron principle". This principle states that the activity and adaptation mechanisms adopted by neurons are aimed at maximizing their impact on the whole network, while at the same time minimizing their power consumption [15]. That is, each neuron seems to be interested in eliciting maximum activity in the surrounding network, since this means maximizing blood flow and thus nutrient and energy supply, while at the same time minimizing its own energy expenditure in producing electrical signals. In short, neurons seem to want to have a positive energy balance. In mathematical terms (generalizing from [14]):

$$E_i(\mathbf{w}_i) = -f(\sum_{k=1} w_{i,k} x_k) + g(\sum_{j=1} x_j) \qquad (1)$$

Here E_i is the energy balance dependent on $\mathbf{w}_i$ the vector of all weights of neuron i, f is the activation function of that neuron, $w_{i,k}$ is the weight for input signal x_k to the neuron and g is function of the output signals of all neurons in the vicinity.

Another critical aspect of these studies is that learning or adaptation is a process that emerges from the behavior of individual neurons. It is a behavior that affects both how each neuron responds to its environment (the surrounding neurons in the network), but also how that neuron interacts with its environment, i.e., how it "injects" its own activity into it. Learning is thus a global dynamic of a network, resulting from the behavior of each of its constituent units.

This auditory case study explores the dynamics of a network of neurons that behave according to the above-mentioned "lazy neuron principle". In this network learning or adaptation to external input signals is the result of the interaction of all neurons in the network and is not controlled and governed by an external function of error minimization.

3.1 Implementation

In this case study, we speculatively simulate the temporal behavior of a recurrent neural network that is constantly adapting to its own dynamics or to some input signal, as in the previous understanding. The neurons are modeled according to the "lazy neuron principle", and learning is understood here as a global process of the whole network that emerges from the behavior of the neurons and is thus not governed by an external optimization process to achieve a predetermined goal. The simulated network is recurrent and consists of a small number of neurons that exchange their output signals. Only one neuron receives the external signal input and the signal produced by one other neuron in the network is taken as the output of the system and audified (see figure 5). We pose that neurons in this network have a defined spatial relationship, and thus, as in biological networks, given the finite conductance of physiological neurons, output signals are delayed differently before being input to other neurons in the network (see figure 6). That is, including the delay factors d_k for the inputs, the above equation 1 becomes:

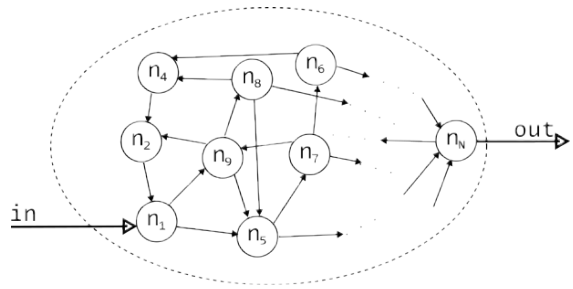


Figure 5. Sketch of the implemented network. Connections in the network are recurrent, the input signal affects one neuron only, while the output signal is read only one other neuron in the network

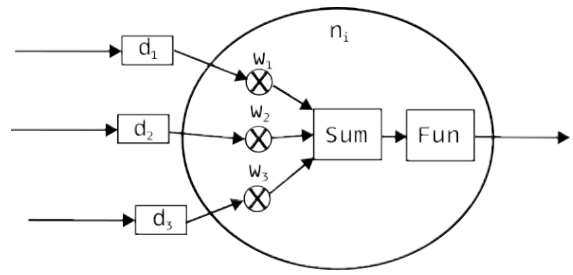


Figure 6. Sketch of the neurons structure in the network. Input signals from other neurons are delayed by a fixed time d_i, multiplied by the respective weight w_i and summed and passed as argument to the activation function.

$$E_i(\mathbf{w}_i) = -f(\sum_{k=1} w_{i,k} x_k(t - d_k)) + g(\sum_{j=1} x_j(t)) \qquad (2)$$

In this case study, neural networks are implemented as dynamical systems composed of many interacting elements. We are not interested in "successful" learning, but rather in the temporal dynamics of the learning system. By simulating the network and directly translating its output into sound, we aim to make its behavior audible. Furthermore, since we are not trying to find the best parameters

for a given task, we keep most of the system's parameters flexible, allowing them to be modulated even while the system is running. Thus, the space of behaviors formulating this specific understanding of learning and adaptation is exposed to exploration or composition and also performance as a sound synthesis instrument. Learning rates, delays, and even the shape of activation functions are preserved as real-time parameters.

The departure of this implementation is thus the maximization of the energy balance function E_i of each neuron i (see equation 1) with respect to that neuron's weights $w_{i,j}$. We formulate this process of gradient ascent in differential form as:

$$w'_{i,j}(t) = +\lambda \frac{\partial E_i(\mathbf{w}_i)}{\partial w_{i,j}} \tag{3}$$

Here, $w'_{i,j}(t)$ is the time derivative of the weight $w_{i,j}$ of neuron i, and λ is the so-called *learning rate*. The sign is positive in this case, as the system tries to maximize its energy balance. In typical backpropagation processes, the exact mathematical form of this derivative is determined prior to the start of the process. In this case study, however, since the choice of the particular form of the activation function is left as a parameter of the system, and since the dependence of the input signals on the weights of each neuron is not known beforehand, an exact mathematical form of the function to be maximized by ascent cannot be derived *a priori*. Thus, we have to find another way to numerically compute the gradient ascent.

As we focus on the temporal dynamics of the system, we compute the gradient of the energy balance in equation 3 using the temporal delays of the weights. The partial derivative in equation 3 can be approximated numerically with

$$\frac{\partial E(\mathbf{w}_i)}{\partial w_{i,j}} = \frac{E(\mathbf{w}_i | w_{i,j}(t-1)) - E(\mathbf{w}_i)}{w_{i,j}(t-1) - w_{i,j}(t)} \tag{4}$$

Where $\mathbf{w}_i | w_{i,j}(t-1)$ is the vector of all weights for neuron i, where only it j-th has taken one step (audio sample) before the current one. It can be seen that the above system works very well in finding the local minimum for the function E as long as the difference $w_{i,j}(t-1) - w_{i,j}(t)$ is not 0. This means that, if the weight $w_{i,j}$ has converged and therefore the above difference is zero, the weifth will not adapt anymore and move to a new minimum when the function or the input signals change.

To solve this problem, we introduce an additional slow dynamics into the optimizing system, which induces very slow (with respect to the dynamics of the input signals) and small (with respect to the values of the weights) oscillations around the local minimum. That is, we made the system third-order and introduced a small limit cycle:

$$\begin{aligned} w'_i(n) =& \lambda v_i + s(r - \sqrt{(w_i - u_i)^2 + v_i^2})w_i \\ v'_i(n) =& -\lambda w_i + \frac{\partial E(\mathbf{w}_i)}{\partial w_{i,j}} \\ & + s(r - \sqrt{(w_i - u_i)^2 + v_i^2})v_i - a v_i \\ u'_i(n) =& w_i - u_i \end{aligned} \tag{5}$$

Where r is the radius of the limit cycle (typically $r \ll w_{i,j}$), s is the strength of the limit cycle attractor, and the u_i component of the dynamical system acts as a leaky integrator of the w_i dynamics. The resulting behavior of the weights w_i is to converge to the local minimum of the function f with rapidly decreasing oscillations and then remain in a kind of *hovering* behavior, slowly oscillating around the converged point. In a way, this hovering behavior allows the neuron to continuously "sense" variations in its inputs and its environment and whenever there is a change in either be able to react and adjust its weights accordingly, finding a new optimal configuration.

This system was programmed and simulated using the *henri* [5] programming language developed by David Pirrò. *henri* is an experimental programming language, developed using the LLVM compiler infrastructure, [6] and originates mainly from the programmer's artistic practice. *henri* is a programming language designed specifically for the formulation and execution of dynamical systems (in the form of systems of ordinary differential equations) using a variable-order symplectic integrator ([16,17]). Dynamical systems programmed in *henri* are simulated (integrated) in audio rate and thus can be used directly for sound synthesis.

3.2 Examples and Artistic Experiments

The networks developed in this case study have been used to synthesize sound in live-electronics performance settings. Each simulated network runs in real-time, and the temporal dynamics of the continuously changing outputs of their neurons are directly used to synthesize sound. Implemented with *henri*, the temporal behavior of the network is directly made audible, while it is still possible to influence the network parameters in real-time during the simulation. This allows the system to be used as a generative sound synthesis engine. The parameters exposed to external interaction, rather than eliciting immediate responses, can influence the system's temporal behavior at different scales. Choosing and exploring different learning rates for the neuron's own energy consumption, the influx of surrounding neuron energies, the attraction of the limit cycle part, the "friction" value, and the shape of the activation functions are part of the performance.

The networks realized so far consist of a small number of neurons, typically 16 to 32, with a small number of connections each (e.g., 3 to 5). The delays used are either randomly chosen or modified while the simulation is running, with durations close to 50 milliseconds. Some sound

[5] `https://git.iem.at/davidpirro/henri`
[6] `https://llvm.org/`

examples are collected on the public web page accompanying this paper. [7] In addition, the *henri* code (a *C* version is planned) used to generate these examples is also made available. [8]

Engaging with these networks requires navigating a complex landscape of behaviors, many of which are challenging to control or "tame." It involves dedicating time to explore subtle changes in parameters and allowing the system to settle into a particular mode. During interaction, these networks exhibit a form of "resistance," a distinct agency of the system that is readily perceptible, making the interaction even more sensible.

With some parameter sets, networks can enter complex self-oscillatory behavior, with different spectral patterns and temporal articulations. These depend strongly on the number of simulated neurons, the chosen activation and energy functions (see equation 1), and the selected delay values. At the same time, the network is open to being influenced by or adapting to external input signals. External sound sources, such as microphone signals or sounds produced by other performers, can be used as input to neurons in the system, and the adaptation behavior to these signals can be modified. Exploring the threshold between a "solipsistic" mode of self-oscillation and a more reactive, input-dependent mode has proven to be one of the most rewarding areas of behavior to explore during performance. The ability to both produce individual sounds and be influenced by external signals is aesthetically and musically inspiring.

As a counterpart to the exploratory attitude of the performative setting, one of the next artistic case studies involves realizing a sound installation for a public space using multiple boards with simple STM32 microcontrollers, [9] a microphone, and a small loudspeaker. Each unit will embody a simplified instance of a neuron in a network, connected through their microphones.

4. CONCLUSIONS

In this paper, we have presented innovative approaches to sound synthesis that leverage the adaptive dynamics of machine learning, not as a means to an end but as an integral part of the creative process. Through the methodological approach of the Speculative Sound Synthesis project, we have explored the aesthetic and performative potentials of speculative sound models, challenging traditional notions of instrumentality and control in computer music. We have described two experiment machine learning systems developed in ways that that diverge from conventional paradigms of neural network applications. The AutoRegNN model demonstrates the richness of incorporating the learning process into sound generation, allowing for real-time, dynamic interactions between the algorithm and its output. The second model described follows the "lazy neuron principle," emphasizing the global dynamics of learning and adaptation as emergent from individual neuron interactions, rather than being driven by external optimization or error minimization.

Our artistic experiments highlight the vast possibilities for future research and artistic exploration, underscoring the significance of small data approaches and the reappropriation of technology for creative expression. These findings open new pathways for understanding and utilizing machine learning in sound synthesis, emphasizing continuous interaction, adaptation, and the exploration of computational materiality as musical and artistic material.

Acknowledgments

This project is funded by the Austrian Science Fund (FWF) within the Programme for Arts-based Research (PEEK) – PEEK AR 713-G.

5. REFERENCES

[1] K. Schulze-Forster, G. Richard, L. Kelley, C. S. J. Doire, and R. Badeau, "Unsupervised music source separation using differentiable parametric source models," 2022. [Online]. Available: https://arxiv.org/abs/2201.09592

[2] F. Ganis, E. F. Knudesn, S. V. K. Lyster, R. Otterbein, D. Südholt, and C. Erkut, "Real-time timbre transfer and sound synthesis using ddsp," 2021. [Online]. Available: https://arxiv.org/abs/2103.07220

[3] Y. Hono, K. Hashimoto, K. Oura, Y. Nankaku, and K. Tokuda, "Sinsy: A deep neural network-based singing voice synthesis system," 2021. [Online]. Available: https://arxiv.org/abs/2108.02776

[4] J. Engel, L. Hantrakul, C. Gu, and A. Roberts, "Ddsp: Differentiable digital signal processing," 2020. [Online]. Available: https://arxiv.org/abs/2001.04643

[5] M. J. Yee-King, L. Fedden, and M. d'Inverno, "Automatic programming of vst sound synthesizers using deep networks and other techniques," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, pp. 150–159, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:4326127

[6] A. Caillon and P. Esling, "Rave: A variational autoencoder for fast and high-quality neural audio synthesis," 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[7] K. Tatar, D. Bisig, and P. Pasquier, "Introducing latent timbre synthesis," 2020. [Online]. Available: https://arxiv.org/abs/2006.00408

[8] C. Kiefer, "Sample-level sound synthesis with recurrent neural networks and conceptors," *PeerJ Computer Science*, vol. 5, p. e205, Jul. 2019. [Online]. Available: http://dx.doi.org/10.7717/peerj-cs.205

[9] J. Impett, "Intelligent company: Co-creative ai as anamnesis," in *Choreomata*, R. A. Trillo and M. Poliks, Eds. Chapman and Hall/CRC, 2023, pp. 219–239.

[7] https://speculative.iem.at/docs/speculations/Experiments_thoughts/unlearn/

[8] https://git.iem.at/davidpirro/unlearn

[9] https://www.st.com

[10] A. v. d. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," 2016. [Online]. Available: https://arxiv.org/abs/1609.03499

[11] J. Benda, "Neural adaptation," *Current Biology*, vol. 31, no. 3, pp. R110–R116, 2021.

[12] W. Gerstner, W. M. Kistler, R. Naud, and L. Paninski, *Neuronal dynamics: From single neurons to networks and models of cognition*. Cambridge University Press, 2014.

[13] J. A. Martín H, J. de Lope, and D. Maravall, "Adaptation, anticipation and rationality in natural and artificial systems: computational paradigms mimicking nature," *Natural Computing*, vol. 8, pp. 757–775, 2009.

[14] A. Luczak, B. L. McNaughton, and Y. Kubo, "Neurons learn by predicting future activity," *Nature machine intelligence*, vol. 4, no. 1, pp. 62–72, 2022.

[15] A. Luczak and Y. Kubo, "Predictive neuronal adaptation as a basis for consciousness," *Frontiers in Systems Neuroscience*, vol. 15, p. 767461, 2022.

[16] E. Hairer, M. Hochbruck, A. Iserles, and C. Lubich, "Geometric numerical integration," *Oberwolfach Reports*, vol. 3, no. 1, pp. 805–882, 2006.

[17] G. Strang, "On the construction and comparison of difference schemes," *SIAM journal on numerical analysis*, vol. 5, no. 3, pp. 506–517, 1968.

UNVEILING THE TIMBRE LANDSCAPE: A LAYERED ANALYSIS OF TENOR SAXOPHONE IN RAVE MODELS

Nádia CARVALHO (nscarvalho@fe.up.pt) (0000-0001-6882-5138) [1],
Jorge SOUSA (jorgevidalsousa@ua.pt) (0009-0004-5920-2448) [2],
Gilberto BERNARDES (gba@fe.up.pt) (0000-0003-3884-2687) [1], and
Henrique PORTOVEDO (henriqueportovedo@ua.pt) (0000-0003-1258-5459) [2]

[1] **University of Porto – Faculty of Engineering and INESC TEC**, Porto, Portugal
[2] **University of Aveiro and INET-md**, Aveiro, Portugal

ABSTRACT

This paper presents a comprehensive investigation into the explainability and creative affordances derived from navigating a latent space generated by Realtime Audio Variational AutoEncoder (RAVE) models. We delve into the intricate layers of the RAVE model's encoder and decoder outputs by leveraging a novel timbre latent space that captures micro-timbral variations from a wide range of saxophone extended techniques. Our analysis dissects each layer's output independently, shedding light on the distinct transformations and representations occurring at different stages of the encoding and decoding processes and their sensitivity to a spectrum of low-to-high-level musical attributes. Remarkably, our findings reveal consistent patterns across various models, with the first layer consistently capturing changes in dynamics while remaining insensitive to pitch or register alterations. By meticulously examining and comparing layer outputs, we elucidate the underlying mechanisms governing saxophone timbre representation within the RAVE framework. These insights not only deepen our understanding of neural network behavior but also offer valuable contributions to the broader fields of music informatics and audio signal processing, ultimately enhancing the degree of transparency and control in co-creative practices within deep learning music frameworks.

1. INTRODUCTION

The intersection of music, deep learning, and creative exploration has spurred transformative advancements in understanding and manipulating sound, offering tools for creating immersive soundscapes [1], transforming existing music [2–4], and synthesizing entirely new sounds [5–7].

WaveNet [8] and SampleRNN [9] were among the pioneering deep-learning approaches for audio modeling in its raw waveform, but their reliance on extensive data and parameters often led to slow synthesis and error accumulation due to their autoregressive nature. Building upon WaveNet, Engel et al. introduced NSynth [5] to tackle representation learning. Additionally, Kumar et al. [10] proposed leveraging generative adversarial modeling to address parallel audio modeling. In the realm of real-time audio processing tasks, a significant milestone is reached with Realtime Audio Variational AutoEncoder (RAVE) models [4]. RAVE models leverage variational autoencoders' capacity to learn audio data representations, enabling dynamic manipulation and synthesis in real-time, and the integration of AI-driven sound design into real-time production workflows [11], such as live performances [12–14] to interactive installations [15].

While these models excel at learning the structures of the audio data, they often remain opaque to users, hindering interaction and control [16]. Navigating RAVE's latent space proves challenging due to the complex and highly-dimensional nature of sound. Nevertheless, transparent representations in audio latent spaces are indispensable for advancing music generation. They offer insight into how the model interprets musical timbres, enabling users to refine outputs and fostering collaboration and innovation. A transparent latent space encourages diverse contributions to model development.

Moreover, direct interaction with musical attributes enhances user engagement and customization. Ethical concerns underscore the need for transparency to manage risks and ensure accountability. Efforts to improve the accessibility of latent space representations encompass various strategies. Intuitive visualization tools and detailed documentation elucidate the space's structure and meaning, empowering users to navigate it effectively. Democratizing access to transparent latent space representations unleashes the full potential of generative models like RAVE, nurturing creativity, collaboration, and innovation in music composition and synthesis.

This paper undertakes a thorough investigation of RAVE models with the objective of elucidating the characteristics of their latent spaces. A particular focus is directed towards a specialized timbre latent space (timbre is understood as all auditory sensations other than pitch, loudness, and perceived duration [17]), meticulously designed to capture the intricate nuances inherent in saxophone extended techniques. We explore the layers of RAVE's encoder and decoder outputs, revealing the transformations and representations at each level through a dual evalua-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

tion approach: 1) a perceptual subjective assessment involving a detailed analysis of the outputs, layer by layer, and comparison across different models trained on diverse datasets; and 2) an objective examination of the extent to which the models' encoding's latent spaces recognize semantically meaningful timbre attributes. This examination is conducted by quantifying intra- and inter-segment distances per technique, register, dynamic, and their combinations across all sounds in the provided dataset. Our evaluation aims to elucidate the fundamental mechanisms governing tenor saxophone timbre representation within the RAVE framework.

Our paper is organized as follows. Section 2 introduces the pre-trained models used for comparison and the datasets employed for their training. Additionally, we outline our methodology for curating a comprehensive collection of sounds representing the entirety of the tenor saxophone dataset, encapsulating the timbre latent space. Section 3 delineates the methodology utilized to evaluate the transparency derived from navigating a latent space generated by RAVE models. This section offers detailed insights into the twofold experimental setup. Subsequently, Section 4 presents and discusses the results obtained. Finally, Section 5 summarizes the conclusions drawn from our study and proposes potential avenues for future research.

2. MATERIALS AND METHODS

Our experimental methods utilize deep learning models (i.e., as Variational Autoencoders) designed to encode timbral information from audio into a latent space. Additionally, we introduce a novel timbre dataset specifically created to capture the subtle timbral variations produced by diverse extended tenor saxophone techniques.

2.1 RAVE models

Realtime Audio Variational AutoEncoder (RAVE) models are deep learning models specifically crafted for generating high-quality audio in real-time [4]. Unlike traditional Variational Autoencoders (VAEs), which can struggle with reconstruction accuracy, RAVE uses a twofold training process (representation learning and adversarial fine-tuning) for enhanced performance. It breaks down audio into multiple frequency bands, capturing fine degrees of detail from large ($\approx$ 48kHz) sampling rate input audio representations without sacrificing efficiency. RAVE models can be used in timbre transfer (i.e., change the timbre of a sound from, e.g., a piano into a violin), for manipulating audio in real-time for performances, or even for compressing audio files.

We begin our exploration with pre-trained RAVE models, detailed in Table 1, originating from both the Institut de Recherche et Coordination Acoustique/Musique (IRCAM) and the Intelligent Instruments Lab (IIL). These models are trained on various raw waveform datasets, as summarized in Table 1, providing a foundation for investigating their capacity to generalize across diverse musical timbres.

The models are built upon different RAVE architectures: IRCAM's models are based on RAVE v2, an enhanced continuous model compared to the one outlined in [4], employing the Variational Auto Encoder objective (ELBO or evidence lower bound function) for regularization. Conversely, IIL's models are trained on a modified iteration of RAVE v1, facilitating transfer learning capabilities. The IIL models exclusively consist of encoder-decoder configurations, without priors, utilizing causal convolutions and optimized for streaming inference with MAX/MSP, Puredata (PD), and Supercollider [18].

Notably, IRCAM's models operate at a sampling rate of 44.1kHz, while IIL's models operate at 48kHz. Furthermore, IIL's models typically feature a greater number of layers, averaging between 16 to 20 layers compared to IRCAM's 8 to 16 layers.

2.2 Tenor Saxophone Timbre Dataset

In this study, our aim was to delve into a timbre latent space, enabling us to capture the nuanced tonal intricacies emanating from the sonic palette of the tenor saxophone. Renowned for its versatility within the timbral spectrum, this instrument offers a plethora of well-documented extended techniques, as documented by Kientzy [25] and Weiss [26]), extensively explored in the context of contemporary Western Art Music.

To document the timbre of the tenor saxophone, we have curated a collection of instrumental tones encompassing a broad array of combinations of (extended) techniques, registers, and dynamics. The details of these combinations are summarized in Table 2. We recorded 191 samples using a stereo pair of microphones (Sontronics STC-1S MT BK) at a sample rate of 48 kHz. The sounds are sampled at 44.1 kHz with a resolution of 24 bits. We provide a more comprehensive list, as well as the dataset, as supplementary material in `https://acesse.one/tenor-saxophone-dataset`.

Extended techniques entail the utilization of the instrument in unconventional manners by expanding the sonic palette [27]. The final set of conventional and extended techniques adopted in our dataset are the following: (1) long non-vibrato notes, and (2) long vibrato notes; (3) long quarter-tone notes; (4) long subtone notes, characterized as an "airy, breathy way of playing lower notes" [26]; (5) flutter-tonguing notes – described as "a kind of tremolo on one tone (...) created with the tip of the tongue" [26]; (6) slap notes – described as "an especially sharp tonguing of a percussive character" [26]; (7) bisbigliando – interpreted as a color trill; (8) multiphonics, resulting from two or more simultaneous pitches sounding on what is otherwise a monophonic instrument; and (9) growling long notes – simultaneous playing and singing.

Due to the unique characteristics of each technique, we accounted for variations in register and dynamics. For instance, in the case of long notes, both with and without vibrato, and quarter-tone long notes, we concentrated on the four different pitch registers previously described – Altissimo, High, Medium, and Low – including two notes for each of these registers: C7 and G6 (altissimo), F6 and C6,(high), D5 and A4 (medium), and C4 and Bb3 (low). [1]

[1] These notations are specifically tailored for tenor saxophone tuned to

Name	Team	RAVE	Sample Rate	Lat. Dim.	Dataset	Ref.
Sol_full	IRCAM	v2	44.1kHz	8	Full Studio OnLine IRCAM database, incorporating over 20,000 samples of extended instrumental playing techniques	[19, 20]
Isis	IRCAM	v2	44.1kHz	8	ISiS Database, comprising synthesized high-quality singing voices, akin to real singers	[21]
MusicNet	IRCAM	v2	44.1kHz	16	Musicnet database, comprising recordings spanning 34 hours of chamber music performances, showcasing 11 instruments	[22, 23]
IILGuitarTimbre	IIL	v1	48kHz	16	Timbre-oriented collection of plucking, strumming, striking, scraping, and more recorded dry from an electric guitar	[18]
SaxSoprano	IIL	v1	48kHz	20	Original soprano saxophone improvisation by Franziska Schroeder	[18]
VocalSet	IIL	v1	48kHz	16	VocalSet dataset, comprehending high-quality recordings of 20 professional singers exhibiting diverse vocal techniques across various vowels	[18, 24]

Table 1. Pre-trained RAVE models employed in our experiments, their characteristics (version, sample rate, and number of latent dimensions), and dataset description.

Technique	Register	Dynamic
Long non-vibrato notes	Altissimo	Very Loud
Long vibrato note	High	Loud
Quarter-Tones	Medium	Medium
Flutter-tonguing	Low	Soft
Slap notes	Very Soft	Very Soft
Growling		
Bisbigliando		
Subtones		
Multiphonics		

Table 2. Collection of sounds encompassing the entire spectrum of the tenor saxophone (timbre space).

Simultaneously, each note was performed at different dynamic levels – Very Loud, Loud, Medium, Soft, and Very Soft.

Given the specific characteristics of the technique, long subtone notes are exclusively applied to the low register of the instrument, specifically notes Low D4, Low C4, Low B3, and Low Bb3. Due to their challenging execution and infrequent usage, flutter-tonguing notes are omitted from the altissimo register (above G6). Additionally, the two notes of each High, Medium, and Low register are studied without dynamic variation.

Efficiency constraints similarly result in the omission of slap notes from the altissimo register. Various registers (High, Medium, Low) were incorporated, each with two notes, featuring variations between a drier attack (secco slap) and a standard attack with clear pitch recognition. Regarding bisbigliando, we established position variations for the established notes for each register (Altissimo, High, Medium, and Low).

In the case of multiphonics, we recorded the following five multiphonics [26, 28] addressing different notes and registers without introducing dynamic variations:

1. C#5, D6 (quarter-tone higher), A6 [2]
2. D5 (three quarter-tone higher), F#5 [3]
3. F#4, G5 [4]
4. D4 (quarter-tone lower), Bb5, D6 (quarter-tone higher), F#6 [5]
5. D5 (quarter-tone higher), G5 (quarter-tone higher), E6, B6 [6]

For Growling Long Notes, we gathered notes distributed across various registers (Altissimo, High, Medium, and Low) without dynamic variations. This decision was made due to the inherent performance limitations of this technique, particularly in soft or very soft dynamics.

These sounds represent a diverse range of timbral qualities and extended techniques inherent to the tenor saxophone, offering potential avenues for exploring the timbre latent space.

3. EVALUATION METHODOLOGY

To explore the transparency afforded by navigating the latent space created by RAVE models within the timbre

B-flat. In concert pitch, the corresponding notes are altissimo (Bb5 and F5), high (Eb5 and Bb4), medium (C4 and G3), and low (Bb2 and Ab2).

[2] concert pitch: B3, C5 (quarter-tone higher), G5

[3] concert pitch: C4 (three quarter-tone higher), E4

[4] concert pitch: E3, F4

[5] concert pitch: C3 (quarter-tone lower), Ab4, C5 (quarter-tone higher), E5

[6] concert pitch: C4 (quarter-tone higher), F4 (quarter-tone higher), D5, A5

domain of the tenor saxophone, we adopt a two-pronged strategy. Firstly, a subjective perceptual assessment aims to offer a comprehensive understanding of the sensitivity of RAVE model layers to various sonic attributes within the context of the tenor saxophone's timbre characteristics. Secondly, an objective evaluation utilizes clustering metrics to analyze the degree to which layers demonstrate sensitivity to multiple sound attributes. This involves assessing whether layers tend to group specific attributes, indicating a heightened sensitivity towards them.

3.1 Perceptual Subjective Assessment

The first evaluation approach delves into the complex layers of the mentioned RAVE model's encoder and decoder outputs, aiming to serve as a guide for systematic empirical analysis.

The perceptual subjective assessment was conducted by the saxophonist Jorge Sousa, who has thirteen years of professional experience playing the tenor saxophone, seeking to provide an in-depth understanding of the sensitivity of RAVE model layers to multiple sonic attributes within the context of the timbre characteristics of the tenor saxophone. This evaluation strives to unveil the unique transformations and representations at various stages of the encoding and decoding processes.

In this scenario, every sound within the dataset undergoes meticulous examination, with thorough listening sessions conducted at each individual layer of every model enumerated in Table 1. The output of each neuron in every layer was converted into a sound signal and listened to individually. The analysis was realized in the visual multimedia programming language Pure Data (PD), [7] using IRCAM's nn_tilde object for PD. [8]

3.2 Analysis of Timbre Clusters

The second evaluation approach objectively assesses the extent to which the models' latent spaces recognize semantically meaningful musical attributes (i.e., timbre, pitch, and dynamics). This assessment quantifies intra- and inter-segment distances across various parameters, such as technique, register, dynamic, and their combination, across all sounds in the provided dataset.

To this end, we construct the latent space for individual sounds, sampled in segments with 40 milliseconds of duration. Clustering is then partitioned into four primary categories: (1) technique, (2) pitch register, (3) dynamic, and (4) all combined. For the initial two categories, we employ metrics on the sounds both with and without normalization, aiming to ascertain whether the overall inherent dynamics of each technique and register significantly influence the model's capacity to discern the attributes under investigation. For normalization, we use librosa [9] to scale the audio input to have a maximum absolute value of 1, ensuring the signal fits within a standard dynamic range.

We utilize two cluster evaluation metrics, the Davis-Bouldin score (DB) [29] and Dunn index [30], to assess both intra-segment distances within each cluster and inter-cluster distances across the latent spaces. The Davis-Bouldin score and Dunn index evaluate the resulting latent space's ability to produce compact and well-separated sound clusters. A lower Davis-Bouldin score indicates better-separated clusters, while a higher score indicates poorly separated clusters. The Dunn index aims for maximization, with the intra-cluster metric (cluster diameter) calculated as the average Euclidean distance across all cluster segments and the inter-cluster metric determined as the distance between each cluster's nearest neighbors. A higher Dunn index indicates more distinct and tightly packed clusters. In this context, the metric compares which layer better discriminates a certain attribute without considering minimum or maximum expected values.

To enable a more objective analysis across the output of each layer, we implement this method for the entire latent space of each sound within the clusters and every individual layer, supplemented by the findings from perceptual assessments. This thorough evaluation aims to uncover the underlying mechanisms governing the representation of tenor saxophone timbre within the RAVE framework. We make the code available at `https://github.com/NadiaCarvalho/SMC-TimbreLandscape.git`.

4. RESULTS AND DISCUSSION

4.1 Perceptual Subjective Assessment

Observations drawn from subjective perceptual assessments of each layer's distinct responses to the sound samples of the tenor saxophone offer valuable perspectives that we can integrate into our objective evaluation process.

For Model **Sol_full**, Jorge's examination indicates that six of the eight layers exhibit no discernible reaction to the instrument's sound samples: layers two, four, five, six, seven, and eight. Layers one and three consistently show minimal differentiation among different pitches and dynamics.

The **Isis** model displays notable reactivity to louder dynamics (Loud and Very Loud) globally, although it is less efficient in responding to various multiphonics. It shows heightened responsiveness in the lower register of the tenor saxophone, particularly with notes such as Bb3, B3, and C4. [10]

Regarding the model trained on **Musicnet**, layer one demonstrates reactivity to nearly all tenor saxophone sound samples, particularly excelling in discerning and responding to samples with distinctive attacks (articulation in slap). However, layers fourteen, fifteen, and sixteen, especially the last one, prove ineffective at discerning most sounds. Layer five stands out for its ability to modify the pitches of introduced sounds.

Model **IILGuitarTimbre** shows layer one's reactivity to all tenor saxophone sound samples, albeit with minimal

[7] `https://msp.ucsd.edu/software.html`, accessed on March 10, 2024.

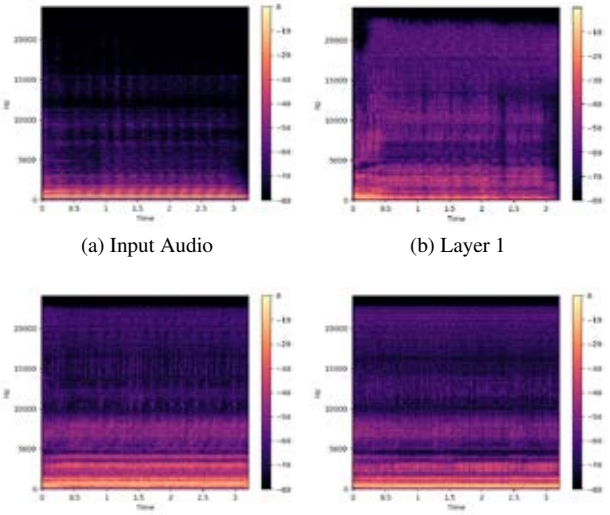
[8] `https://github.com/acids-ircam/nn_tilde`, accessed on March 10, 2024.

[9] `https://librosa.org`, accessed on March 10, 2024.

[10] Pitch in this section is always presented in the notation specifically tailored for tenor saxophone tuned to B-flat.

responsiveness to changes in dynamics and pitch. Conversely, layers two, five, ten, thirteen, and fourteen show no response to any stimulus. Layer seven demonstrates noticeable reactions to sound stimuli at low dynamics (Soft and Very Soft).

The **SaxSoprano** model is generally deemed ineffective. Among its twenty layers, eleven show no reaction to tenor saxophone sound samples. Layer one emerges as the most responsive, consistently reacting across various dynamics and effectively responding to slap articulations and multiphonics. Layer eleven consistently reacts to the note D5 (medium register), a unique trait not found in other models. Activating layer one influences other layers, leading to distinct behaviors, such as amplifying low frequencies, adding granular texture, producing a "Pan flute" sound, and closely resembling the input pitch.



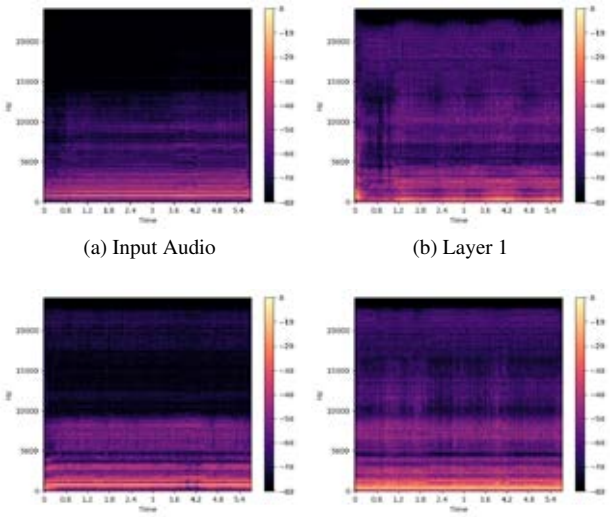
(a) Input Audio (b) Layer 1

(c) Layer 11 (d) Layer 16

Figure 1. Spectrograms for the input audio of the soft low C recording, as well as the sounds decoded from layers 1, 11, and 16 of the latent space.

In our examination of the **VocalSet** model comprising sixteen layers, we found it to be a balanced model, with layers responding to sound samples across all registers (from Low to Altissimo), except for layer sixteen, which showed greater reactivity to the Low register (as observed in Fig. 1). While several layers reacted to multiphonics (see Fig. 2), the second and third multiphonics didn't elicit any response. Layer one emerged as the most reactive layer, layer eleven displayed high reactivity to the Altissimo register, and layer thirteen responded solely to note C across various registers.

Finally, among the models studied, we propose that the Musicnet and Vocalset models are the most reactive to tenor saxophone sound samples. Despite both being saxophones, the model trained on soprano saxophone sounds proves ineffective when examined layer-by-layer. Layer one consistently exhibits the most robust reaction to various registers, pitches, effects, and dynamics. The models demonstrate higher reactivity to louder dynamics (Loud or Very Loud) and are generally more effective in lower



(a) Input Audio (b) Layer 1

(c) Layer 10 (d) Layer 16

Figure 2. Spectrograms for the input audio of the multiphonic number five, as well as the sounds decoded from layers 1, 11, and 16 of the latent space.

notes, particularly Bb3. Additionally, we observe an apparent inability to react to and distinguish between various presented multiphonics.

4.2 Analysis of Timbre Clusters

Table 3 shows the cluster analysis results, encompassing all layers of each model. From these findings, several conclusions emerge. Foremost among them is the clear indication that the absence of normalization negatively impacts the Davis-Bouldin score when applying clustering metrics to extended techniques and pitch registers across all models. Consequently, the models struggle to effectively discern extended techniques, and consequently, timbre changes or register variations when dynamics vary. The exception to this trend is observed in the VocalSet model, where both normalized and non-normalized audio inputs yield very similar values for both technique and pitch register.

Among the models evaluated, the SaxSoprano model stands out for its superior ability to distinguish between extended techniques, particularly when the audio is normalized. With the second-best Davis-Bouldin score and the highest Dunn index score, it demonstrates notable proficiency in this aspect. A noteworthy finding concerns the MusicNet model, which achieves the best Davis-Bouldin score but one of the lowest Dunn index scores. This suggests that while the MusicNet model excels in forming distinct timbre clusters due to its capacity to capture unique data patterns, it struggles to achieve adequate separation between these clusters. Consequently, it faces challenges differentiating characteristics to create clearly distinct timbre clusters.

Regarding pitch registers, most of the models perform similarly with normalized audio. The exception is the IILGuitarTimbre, which presents very high values of the Davis-Bouldin score, meaning it cannot distinguish pitch

Model		Technique		Register		Dynamic	All
		N	NN	N	NN		
Sol_full	**DB**	9,3978	63,1686	3,1654	71,0053	61,2390	23,3833
	Dunn	0,0001	0,0003	0,0002	0,0003	0,0002	0,0014
Isis	**DB**	7,8626	36,9219	**2,5262**	17,9766	30,7508	18,4759
	Dunn	0,00002	0,0003	0,0001	**0,0005**	0,0001	0,0006
MusicNet	**DB**	**6,9871**	27,2322	3,4207	22,0426	35,3139	14,3542
	Dunn	0,00002	0,0003	0,0001	**0,0005**	**0,0003**	0,0016
IILGuitarTimbre	**DB**	46,4832	69,0797	25,9224	40,3061	58,2117	35,3975
	Dunn	0,0002	0,0002	**0,0005**	0,0003	0,0001	0,0003
SaxSoprano	**DB**	8,8365	23,6323	3,1240	5,8060	14,4397	9,2430
	Dunn	**0,0003**	**0,0004**	0,0004	**0,0005**	0,0002	**0,0034**
VocalSet	**DB**	11,4939	**12,7429**	2,5283	**2,9858**	**8,4275**	**5,5918**
	Dunn	**0,0003**	0,0003	0,0002	**0,0005**	**0,0003**	0,0024

Table 3. Clustering Results Across All Layers. The best results for each clustering category and metric are marked in bold. N and NN denote normalized and non-normalized audio, respectively.

well. However, contrary to the previous example regarding extended techniques, it features the best Dunn index. This suggests that the dataset used for training may encompass greater complexity regarding pitch registers associated with extended techniques. This distinction might be attributed to the fact that the IILGuitarTimbre model is trained on a dataset featuring sounds more distant from those of the tenor saxophone, unlike the other models, which are associated with instrumental orchestral and chamber music sounds or voice, often resembling saxophone timbres.

Concerning dynamics (coupled with extended techniques) and all sounds clustered separately, the VocalSet model emerges as the top performer for both metrics, closely followed by the SaxSoprano model.

Figure 3 showcases the outcomes of the cluster analysis, organized layer by layer. Due to the close resemblance of Dunn index results across all metrics (approximately 1E-05), only Davis-Bouldin score results are presented in the figure. Several discernible patterns emerge from these results, complementing the listening test findings. Specifically, dynamics are consistently better generalized in the initial layer of the model, except in the case of the IILGuitarTimbre and VocalSet models, where they are observed in the second and sixth layers, respectively. Combined sounds are not adequately clustered at any model layer, likely due to the limitation of a single layer in capturing the diverse timbre attributes of entire sounds. The pitch register is generally well perceived across the models, although not consistently at the same layer in every model. At the level of extended techniques, Musicnet, closely followed by VocalSet and Isis, are the best models at semantically perceiving timbric changes by a single layer.

5. CONCLUSIONS AND FUTURE WORK

This study conducts a comprehensive examination of RAVE models to uncover the nuances of their latent spaces. Emphasizing a specialized timbre latent space tailored for saxophone extended techniques, we delve into the encoder and decoder outputs of RAVE layers. Through a dual evaluation method, we subjectively analyze outputs layer by layer and compare them across diverse model datasets. Additionally, we objectively assess the models' encoding latent spaces' ability to recognize semantically meaningful timbre attributes. This assessment involves quantifying intra- and inter-segment distances across various sound attributes, including technique, register, and dynamics, across the entire dataset.

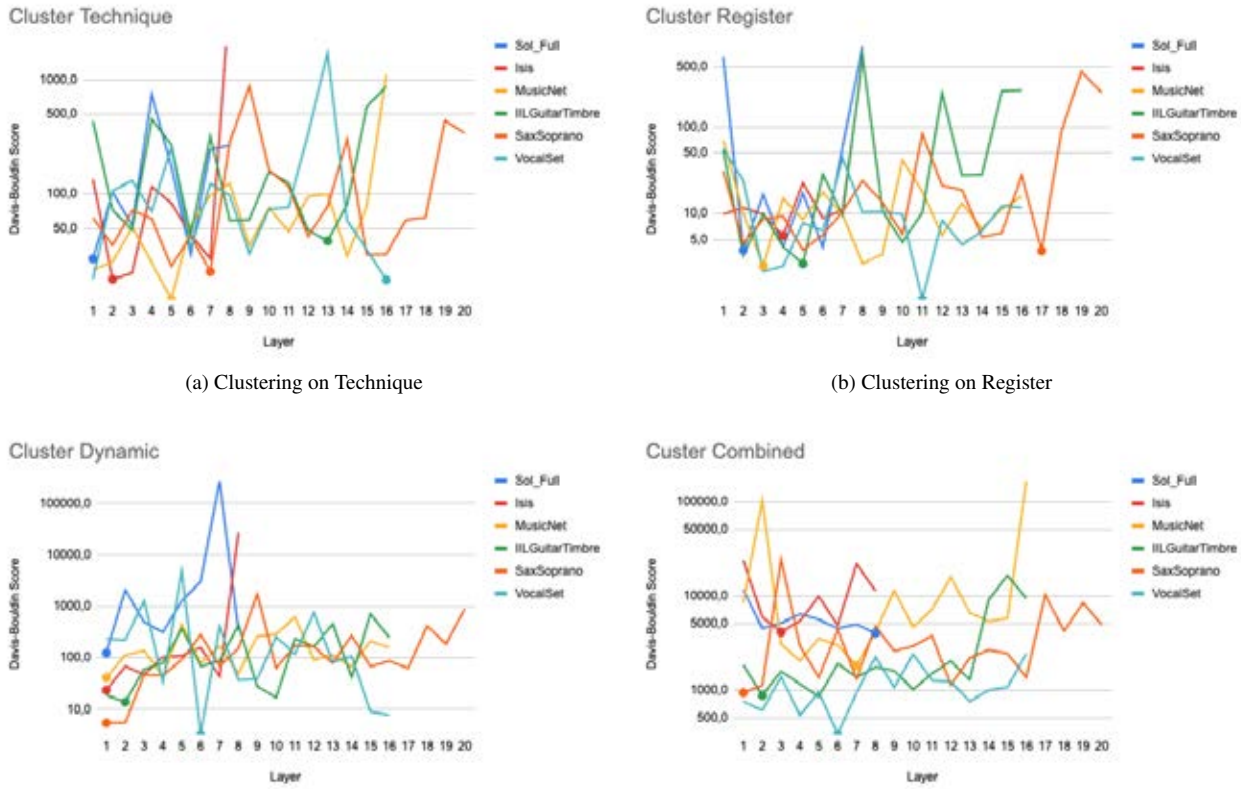
The conclusions drawn from this investigation into RAVE models reveal crucial insights into their ability to capture and represent saxophone timbre attributes. Through a meticulous dual evaluation approach encompassing subjective listening tests and objective clustering analyses, several key findings emerge.

Firstly, among the models studied, MusicNet and VocalSet exhibit heightened reactivity to tenor saxophone samples, particularly evident in the subjective perceptual layer-by-layer examination. However, the effectiveness varies across models, with the soprano saxophone model proving less reactive. Notably, the initial layer consistently demonstrates robust responses to different attributes, highlighting its pivotal role in timbre representation.

Secondly, the absence of normalization negatively impacts clustering metrics, hindering the models' ability to effectively distinguish extended techniques and pitch registers. While some models, such as SaxSoprano, excel in distinguishing between extended techniques, others struggle to achieve adequate separation, as seen in the case of MusicNet. Furthermore, dynamics and combined sounds present additional challenges in clustering, underscoring the complexity of timbre representation.

Lastly, objective layer-by-layer analysis complements the findings from perceptual evaluation, uncovering consistent patterns in dynamics perception across models. While there are variations in pitch register perception across different layers, MusicNet and VocalSet stand out as top performers in semantically perceiving timbral changes within a single layer.

In summary, the comprehensive evaluation elucidates fundamental mechanisms governing tenor saxophone timbre representation in RAVE models. The insights gleaned



(a) Clustering on Technique

(b) Clustering on Register

(c) Clustering on Dynamic

(d) Combined Clustering

Figure 3. Davis-Bouldin Score Results by Layer. The best results are marked with a circle. All audio is normalized for the cluster categories corresponding to extended techniques and pitch registers.

contribute to advancing our understanding of neural network-based timbre modeling and hold implications for music synthesis and analysis applications.

Moving forward, we plan to further investigate the RAVE model layers by analyzing their behavior across individual registers, dynamics, and extended techniques. Such an endeavor promises to unveil further insights into the intricacies of RAVE models, thereby enhancing transparency in their functioning. Additionally, we aspire to move beyond mere transparency and innovate new tools that empower users to harness the potential of these models creatively. By enabling users to take control and experiment with the models, we envision facilitating the creation of music and performances that are enriched and elevated. This forward-looking approach not only advances our understanding of RAVE models but also fosters a dynamic and collaborative environment for music exploration and expression.

Acknowledgments

This research has been funded by the Portuguese National Funding Agency for Science, Research and Technology [2021.05132.BD and 2023.01345.BD].

6. REFERENCES

[1] H. Scurto and A. Chemla-Romeu-Santos, "Deeply Listening Through/Out the Deepscape," in *28th International Symposium on Electronic Art (ISEA 2023)*, Paris, France, May 2023. [Online]. Available: https://hal.science/hal-04108995

[2] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "Timbretron: A wavenet(cyclegan(cqt(audio))) pipeline for musical timbre transfer," *Computing Research Repository (CoRR)*, vol. abs/1811.09620, 2018. [Online]. Available: http://arxiv.org/abs/1811.09620

[3] D. K. Jain, A. Kumar, L. Cai, S. Singhal, and V. Kumar, "Att: Attention-based timbre transfer," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–6.

[4] A. Caillon and P. Esling, "RAVE: A variational autoencoder for fast and high-quality neural audio synthesis," *Computing Research Repository (CoRR)*, vol. abs/2111.05011, 2021. [Online]. Available: https://arxiv.org/abs/2111.05011

[5] J. Engel, C. Resnick, A. Roberts, S. Dieleman, M. Norouzi, D. Eck, and K. Simonyan, "Neural audio synthesis of musical notes with WaveNet autoencoders," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 1068–1077. [Online]. Available: https://proceedings.mlr.press/v70/engel17a.html

[6] F. Roche, T. Hueber, S. Limier, and L. Girin, "Autoencoders for music sound modeling : a comparison of linear, shallow, deep, recurrent and variational models," in *SMC 2019 - 16th Sound & Music Computing Conference*, ser. Proc. of SMC 2019, U. of Malaga (UMA), Ed., no. 1-6, Malaga, Spain, May 2019. [Online]. Available: https://hal.science/hal-02349406

[7] K. Tatar, D. Bisig, and P. Pasquier, "Latent timbre synthesis: Audio-based variational auto-encoders for music composition and sound design applications," *Neural Computing and Applications*, vol. 33, no. 1, p. 67–84, Oct. 2020. [Online]. Available: http://dx.doi.org/10.1007/s00521-020-05424-2

[8] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *Computing Research Repository (CoRR)*, vol. abs/1609.03499, 2016. [Online]. Available: http://arxiv.org/abs/1609.03499

[9] S. Mehri, K. Kumar, I. Gulrajani, R. Kumar, S. Jain, J. Sotelo, A. C. Courville, and Y. Bengio, "Samplernn: An unconditional end-to-end neural audio generation model," *Computing Research Repository (CoRR)*, vol. abs/1612.07837, 2016. [Online]. Available: http://arxiv.org/abs/1612.07837

[10] K. Kumar, R. Kumar, T. de Boissière, L. Gestin, W. Z. Teoh, J. M. R. Sotelo, A. de Brébisson, Y. Bengio, and A. C. Courville, "Melgan: Generative adversarial networks for conditional waveform synthesis," in *Neural Information Processing Systems*, 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1910.06711

[11] S. Nercessian, "P-rave: Improving rave through pitch conditioning and more with application to singing voice conversion," in *Proceedings of the 26th International Conference on Digital Audio Effects (DAFx23)*, 2023.

[12] F. Schroeder and F. Reuben, "Ai Improvised Music Duo," *AIMC 2023*, aug 29 2023, https://aimc2023.pubpub.org/pub/8x9jxz9a.

[13] J. C. Reus, "i o we," *AIMC 2023*, aug 29 2023, https://aimc2023.pubpub.org/pub/hpy32yre.

[14] J. Armitage, "Gagnavera," *AIMC 2023*, aug 29 2023, https://aimc2023.pubpub.org/pub/ppgwfht1.

[15] ——, "Strengjavera," *AIMC 2023*, aug 29 2023, https://aimc2023.pubpub.org/pub/83k6upv8.

[16] A. Chemla–Romeu-Santos and P. Esling, "Challenges in creative generative models for music: a divergence maximization perspective," in *Proceedings of the 3rd Conference on AI Music Creativity*. AIMC, Sep. 2022. [Online]. Available: https://doi.org/10.5281/zenodo.7088272

[17] T. Letowski, "Timbre, tone color, and sound quality: concepts and definitions," *Archives of Acoustics*, vol. 17, pp. 17–30, 1992.

[18] Intelligent Instruments Lab, "rave-models (revision ad15daf)," 2023. [Online]. Available: https://huggingface.co/Intelligent-Instruments-Lab/rave-models

[19] G. Ballet, R. Borghesi, P. Hoffmann, and F. Lévy, "Studio Online 3.0: An Internet "Killer Application" for Remote Access to IRCAM Sounds and Processing tools," in *Journées d'Informatique Musicale*, Paris, France, May 1999. [Online]. Available: https://hal.science/hal-03112091

[20] C. Cella, D. Ghisi, V. Lostanlen, F. Lévy, J. Fineberg, and Y. Maresz, "Orchideasol: a dataset of extended instrumental techniques for computer-aided orchestration," *Computing Research Repository (CoRR)*, vol. abs/2007.00763, 2020. [Online]. Available: https://arxiv.org/abs/2007.00763

[21] L. Ardaillon, "Synthesis and expressive transformation of singing voice," Theses, Université Pierre et Marie Curie - Paris VI, Nov. 2017. [Online]. Available: https://hal.science/tel-01710926

[22] J. Thickstun, Z. Harchaoui, and S. Kakade, "Learning features of music from scratch," 2017.

[23] J. Thickstun, Z. Harchaoui, and S. M. Kakade, "Musicnet," Jul. 2021. [Online]. Available: https://doi.org/10.5281/zenodo.5120004

[24] J. Wilkins, P. Seetharaman, A. Wahl, and B. Pardo, "Vocalset: A singing voice dataset," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, September 23-27, 2018*, E. Gómez, X. H. 0001, E. Humphrey, and E. Benetos, Eds., 2018, pp. 468–474. [Online]. Available: http://ismir2018.ircam.fr/doc/pdfs/114_Paper.pdf

[25] D. Kientzy and D. Charles, "Saxologie du potentiel acoustico - expressif des 7 saxophones," Ph.D. dissertation, Paris 8, 1990.

[26] M. Weiss and G. Netti, *The Techniques of Saxophone Playing*. Bärenreiter, 2010.

[27] M. Burtner, "Making noise: Extended techniques after experimentalism," *New Music Box*, vol. 6, 2005.

[28] D. Kientzy, *Les Sons Multiples aux Saxophones*. Editions Salabert, 1982.

[29] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 224–227, 1979.

[30] J. C. Dunn†, "Well-separated clusters and optimal fuzzy partitions," *Journal of Cybernetics*, vol. 4, no. 1, pp. 95–104, 1974. [Online]. Available: https://doi.org/10.1080/01969727408546059

TEMPO ESTIMATION ON MULTIPLE METRICAL LEVELS WITH SEQUENCY FLUX AND MULTIRESOLUTION ANALYSIS

Savvas KAZAZIS (savvas.kazazis@mail.mcgill.ca) [1,2]

[1] *School of Music Studies*, **Aristotle University of Thessaloniki**, Thessaloniki Greece
[2] *Department of Informatics and Telecommunications*, **National and Kapodistrian University of Athens**, Athens Greece

ABSTRACT

We present a method that accurately estimates multiple tempi simultaneously by acting on different time-scales of the signal. We introduce the term sequency flux, which refers to an audio feature derived from the Walsh-Hadamard transform, as an alternative to the widely used spectral flux. The signal derived from sequency flux is further decomposed using the Empirical Mode Decomposition (EMD) and multiple tempo hypotheses are formed by evaluating the autocorrelation function on the intrinsic mode functions (IMFs) over a range of analysis window lengths. An inference method is then proposed that filters out weak hypotheses. The inference lies on measuring the inter-onset intervals (IOIs) found within each IMF, and on how strongly these correspond to the respective tempo hypotheses. The method is extremely simple to implement yet robust, because it does not rely on complex heuristics, the use of arbitrary and signal dependent thresholds, or learning.

1. INTRODUCTION

Tempo perception and beat induction are fundamental musical traits that allow one to enjoy and appreciate music. It is because of these traits that one can dance to the music, or participate in collaborative music-making activities. As Honing nicely puts it [1]: "Without it no music". It is therefore no surprise that topics related to tempo estimation and beat tracking have received much attention in the music information retrieval community [2]. Most algorithms aim to estimate at least one of the three following metrical levels: the tatum (or temporal atom; [3]), the tactus (or beat) and the measure (or bar). These levels relate to the detection of musical events operating on different time scales. The tatum refers to the shortest durational value of events (e.g., a sixty-fourth note). The tactus is the most prominent level and refers to the tapping-rate that a listener is most likely to tap to while listening to music. Tempo is usually derived from the tactus level. The measure level relates to the rate of harmonic changes, or to the period of rhythmic and melodic patterns.

In the fields of computational ethnomusicology and especially in performance studies, one may be interested in analyzing the morphological structures of musical events that occur on metrical levels other than the tatum, tactus, and measure. Stylistic conventions (related to a particular genre, for example) and aesthetic principles of performers and composers may appear both in macro and micro time scales, other than the measure or tatum levels [4]. However, the vast majority of algorithms aim at estimating just one metrical level, which is often related to tactus (i.e., the tempo). Although one may assume that tempi that are multiples or submultiples of the basic tempo are equally valid, this assumption may not always agree with listeners' perceptions (Section 2). This is because actual events may not be present (or cannot be implied) on hypothetical metrical levels. As Goto mentions in [5], "The principal reason that beat tracking is intrinsically difficult is that it is the problem of inferring an original beat structure that is not expressed explicitly." In addition, systems which detect just one metrical level, may have limited success when applied to musical pieces composed with polyrhythms.

In this work, we propose a method that aims to estimate all the characteristic time scales in which musical events occur. This is accomplished through a decomposition method (i.e., the Empirical Mode Decomposition; Section 3), and the formation of multiple tempo hypotheses that are derived from the resulting components of the decomposition. We are particularly interested in testing whether these components actually carry tempo information on metrical levels that match listeners' perceptions. In the following, we first present related work, and some algorithms that were evaluated in the MIREX campaigns along with results obtained from two of the most popular deep learning approaches (Section 2). In Section 3, we present the core elements of the proposed method, and in Section 4 the method itself along with evaluation results on a limited dataset from the MIREX campaign, which served as a case study. In Section 5 we summarize our conclusions and point to some future directions.

2. RELATED WORK & CHALLENGES

2.1 Onset Detection & Spectral Flux

Most tempo estimation methods start by first applying an onset detection function to the signal in order to estimate the positions of note onsets. The onset detection function can be energy-based or spectral-based. Energy based methods usually consist of the following steps. The audio signal is first rectified (e.g., by squaring it) and then

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

smoothed by a low-pass filter in order to estimate its (power) amplitude envelope. Sudden energy changes in the amplitude envelope are then captured through differentiation. For onset detection, it is usually assumed that only the energy increments observed in the differentiated amplitude envelope are relevant to note onsets and therefore, the signal is half-wave rectified in a subsequent step. The last step involves a peak-picking strategy (e.g., [6, 7]) in order to eliminate spurious peaks, which may not correspond to actual note onsets. Energy-based methods generally provide acceptable results when applied to musical pieces that exhibit strong percussive onsets. However, the results are relatively poor when these methods are applied to musics which mainly consist of soft onsets (e.g., orchestral pieces, string quartets, chorales), or to dense musical passages in which notes occur asynchronously and occupy different frequency bands.

Spectral-based methods [8] offer an improvement over energy-based methods because the differentiation occurs within particular frequency bands of the audio signal, and therefore minimize the possibility of an onset to go unnoticed due to energy masking. The first step in spectral-based methods is the computation of *spectral-flux*, which refers to the computation of the short-time Fourier Transform (STFT) followed by differentiation (across time) over each frequency band. Alternatively, some methods [9] group the frequency bins of the STFT into several bands (e.g., according to Mel-scale) and differentiation take place between those bands instead of the individual frequency bins. Many methods also use the power instead of the magnitude of frequency bands or some other form of logarithmic compression [10] prior to differentiation. The differences are then half-wave rectified and summed across all frequency bands. The last step usually involves some peak-picking strategy as in the energy-based methods.

Dixon [11] compared several different methods related to the computation of spectral flux for the purpose of onset detection, but found that the magnitude-based spectral flux (as described above) performs in par with other more complicated methods (e.g., methods that also take phase into account). Energy fluctuations in a particular spectral band (e.g., tremolo) may lead to the detection of erroneous onsets. Although thresholding through a peak-picking strategy may improve the results, fluctuations in frequency (e.g., vibrato) also cause “blurring” of the peaks in the final onset detection function. A solution to this problem was proposed by Böck et al. [12] through an onset detection algorithm with vibrato and tremolo suppression ("Superflux"), which reduces the number of false-positive onset detections for musical pieces that exhibit strong vibrato. However, it should be noted that the reduction of false positive onsets through some form of suppression require the use of carefully chosen thresholds and that the optimal values of which may vary across musical genres, or instrumentation.

2.2 Periodicity Analysis & Tempo Estimation

Tempo estimation usually relies on finding the dominant periodicity of the onset detection function. One of the most commonly used methods in detecting such periodicities is the autocorrelation function (e.g., [13, 14]). Other methods include the Fourier Transform [10] and comb filterbanks [9, 15] or a bank of resonators applied to chroma features derived from the harmonic and percussive components of the signal [16]. Peeters [17] explores the fact that octave uncertainties present in Fourier-based methods and autocorrelation-based methods occur in inverse domains (i.e., in frequency domain and time-lag-domain, respectively) and proposes a combination of the two. Another group of methods detect dominant periodicities directly from the inter-onset intervals (IOIs) found in the onset detection function through histogramming [18], or some clustering scheme [19]. Many tempo estimation systems are also equipped with an induction algorithm [19] in order to make more accurate predictions given a set of tempi candidates, which possibly correspond to different metrical levels (e.g., tatum, tactus). Common algorithms used for this task include the use of Hidden Markov Models (HMMs; e.g., [9,17]) and dynamic programming (e.g., [16, 20]).

In relation to modelling multiple hierarchical metrical levels, one of the first systems was proposed by Goto [5]. The system potentially recognizes hierarchical beat structures consisting of the quarter note level, the half-note level, and the measure-level. The latter is computed under the assumption that the time signature of the musical track is 4/4. In relation to quarter note level (i.e., the beat level) the system assumes that the tempo range is between 61 BPM and 185 BPM for drum tracks, or between 61 BPM and 120 BPM for music without drums. Klappuri et al. [9] proposed a method that simultaneously estimates the metrical levels of tatum, tactus (i.e., the beat), and measure. The HMM is designed to take into account temporal dependencies between successive estimates of each metrical level, and imposes explicit dependencies between the periods and phases of the tatum, tactus and measure levels. Lartillot et al. [21] present a method that tracks hierarchical metrical structures at various levels expressed as a set of detected periodicities in pairwise harmonic relations. This method requires no training (compared to the method presented in [9], for example) and relies on signal processing (e.g., by using methods similar to spectral flux and autocorrelation) as well as heuristics-based peak tracking.

Deep learning approaches to tempo estimation include the models of [22] and [23]. In addition to being widely cited and having publicly available open-source implementations, these methods are also considered to be some of the most accurate in relation to particular datasets. Table 1 provides the evaluation results of these two approaches, and the rest of the methods mentioned above. These results stem from the MIREX tempo estimation campaigns in which the task was to estimate two tempi correctly along with their perceptual salience (see also Section 4).

From this table we can see that the deep learning approaches perform better than the other methods in estimating at least one tempo correctly (although perhaps in some cases with questionable statistical significance). However, their accuracy in estimating correctly a second tempo is

Reference	[22]	[23]	[16]
One tempo correct	0.99	0.98	0.94
Both tempi correct	0.69	0.66	0.62
Reference	[9]	[20]	[21]
One tempo correct	0.94	0.93	0.92
Both tempi correct	0.61	0.46	0.57

Table 1. Evaluation results of some algorithms submitted to the MIREX campaigns over the years. (Results from [21].)

relatively low. Interestingly, methods designed for modelling multiple hierarchical levels perform less well in this task compared to other generic methods. These results are rather discouraging in relation to the relatively high 8% tolerance level, which was used in the MIREX campaigns, instead of the widely accepted 4% tolerance level and according to which Accuracy 1 (*Acc1*) is measured.

Another popular and relatively accurate deep learning approach for simultaneously tracking tempo, beat, and downbeat is proposed by Böck et al. [24]. While this method estimates a single tempo (based on datasets with tempo and/or beat tracking data), an additional tempo hypothesis could potentially be formed through the estimation of downbeats. However, downbeats do not always directly relate to the measure level. Table 2 lists the evaluation results of this method on tempo estimation, alongside some previously mentioned methods on specific datasets, as reported in [24]. Comparing these results to those for secondary tempo estimation in Table 1, it can be inferred that nearly perfect Accuracy 2 (*Acc2*) scores rarely reflect listeners' perceptions of tempo.

Reference	[24]	[23]	[16]	[13]
Lowest *Acc1*	0.830	0.769	0.651	0.506
Highest *Acc1*	0.870	0.821	0.725	0.733
Lowest *Acc2*	0.950	0.926	0.922	0.924
Highest *Acc2*	0.990	0.976	0.979	0.972

Table 2. Evaluation results of the method presented in [24] compared to other algorithms. "Lowest *Acc*" and "Highest *Acc*" represent the minimum and maximum accuracy scores observed across the datasets used in the study. (Results from [24].)

3. SEQUENCY FLUX & THE EMPIRICAL MODE DECOMPOSITION

3.1 The Walsh-Hadamard Transform & Sequency Flux

It is well known that narrow-band signals that possess harmonic structures such as voiced speech, or sustained and pitched instrument sounds, can be well represented by the FFT through a small number of harmonic complex exponentials. Abruptly changing signals, such as percussive onsets, unvoiced speech, or complex musical passages, are often random-like, "noisy", and require a large number of Fourier coefficients in order to be accurately represented, and therefore may not be characterized efficiently. Such broadband signals can be characterized more efficiently by a broadband class of basis functions such as the Walsh functions [25]. The Walsh-Hadamard transform is analogous to the Fourier transform, but instead of sinusoids it consists of a set of periodic and aperiodic rectangular waves that take only two amplitude values of +1 and -1. These waves are characterized by their sequency (as opposed to frequency in the Fourier transform), which denotes the number of zero-crossings the function makes per unit time.

Hadamard matrices of order $M = 2^k$ can be generated recursively from the following relationship [26]:

$$\mathbf{H}(k+1) = \begin{bmatrix} \mathbf{H}(k) & \mathbf{H}(k) \\ \mathbf{H}(k) & -\mathbf{H}(k) \end{bmatrix}, \quad k = 0, 1, 2, ..., \quad (1)$$

with $H(0) = 1$.The Walsh-Hadamard transform of a real valued signal $\mathbf{x}$ of length M can then be computed from:

$$\mathbf{X} = \mathbf{x} \cdot \mathbf{H}^{(M)}, \quad (2)$$

where the coefficients of $\mathbf{X}$ correspond to the weights of the Walsh functions.

The Walsh-Hadamard transform can be applied sequentially to (possibly overlapping) frames of x in a similar way to the short-term Fourier Transform (STFT) with the only minor restriction being that the frame size should be a power of two. If we let $X(n, s)$ represent the *s*th sequency of the *n*th frame, then the computation of sequency flux ($SeqFlux$) becomes identical to the computation of (magnitude-based) spectral flux:

$$SeqFlux[n] = \sum_{s=1}^{S} |X(n,s)| - |X(n-1,s)| \quad (3)$$

3.2 The Empirical Mode Decomposition (EMD)

The Empirical Mode Decomposition (EMD) [27] is a non-linear and adaptive method for analyzing non-stationary signals and data. With this method, the signal is represented as a sum of zero-mean amplitude and frequency modulated (AM/FM) components. In the context of EMD, these components are called Intrinsic Mode Functions (IMFs) and (in theory) must satisfy the following two conditions: (i) the number of extrema (i.e., the local maxima and local minima of the signal) and the number of zero crossings must be equal, or differ at most by one; and (ii) the envelopes defined by local maxima and local minima must be symmetric, or in other words, the mean value of the upper and lower envelopes must be zero.

EMD is based on an iterative process that successively removes the fine structure of the signal through some form of adaptive and signal-dependent time variant filtering. The IMFs are computed according to the following steps [28]:

1. Estimate all the maxima and minima of the signal $x(t)$

2. Construct the upper ($u(t)$) and lower ($l(t)$) envelopes of the signal by interpolating the maxima and minima, respectively
3. Compute the mean envelope $m(t) = (u(t)+l(t))/2$
4. Extract the detail $d(t) = x(t) - m(t)$
5. Iterate from step (1) to (4) on $d(t)$ until it can be considered zero-mean (and therefore an IMF)
6. Set $d(t)$ to be an IMF, $i(t) = d(t)$
7. Repeat from the beginning on the residual, $r(t) = x(t) - i(t)$ as the data
8. Stop the process if $r(t)$ remains approximately constant, or is monotonic

The procedure of repeated iterations that occur in step 5 is known as *sifting*. The algorithm will eventually converge, because the number of extrema is decreased when progressing from one residual to the next.

To the best of authors' knowledge, there are only a few approaches for tempo estimation using the EMD algorithm [29, 30]. In [29], Pikrakis et al. used EMD to segment music recordings into regions that exhibit similar rhythmic characteristics, and to analyze the diagonals of the self-similarity matrix of those regions in order to extract the tempo. The method presented in [30] is conceptually similar to the one we present here, and therefore merits some discussion. This method computes IMFs of the input signal, and estimates their corresponding amplitude envelopes using low-pass filtering and half-wave rectification. The periodicities of each IMF are estimated through the autocorrelation function (ACF). Tempo induction is made in two steps. In the first step, the IMFs that do not exhibit harmonically related peaks in the respective ACF are removed. In the second step, the induction algorithm searches for one-to-one (peak) correspondences and harmonic relations (in terms of time-lags) between the ACF peaks of a particular IMF in relation to subsequent ones. If no match is found, the induction method fails to evaluate the tempo. Although this method is conceptually similar to the one presented in this paper, it is fundamentally different to the one proposed in this work and which we detail in the next section.

4. MULTIRESOLUTION TEMPO ESTIMATION

4.1 Formation of Multiple Tempo Hypotheses

The method starts with a pre-processing stage that includes the following steps: conversion to mono by selecting the channel with the highest root mean square amplitude; resampling to 22.5 kHz; and removal of the global DC offset. After this stage, the Walsh-Hadamard transform is computed using a window length of 512 samples and a hop-size of 256 samples, which corresponds to about 11.6 ms. The sequency flux is then computed according to Equation (3). We deliberately avoid the use of half-wave rectification (HWR) on sequency flux, because we assume that offsets also play a role in tempo perception [1]. To achieve a better resolution, the frame rate of sequency flux gets upsampled to 200 Hz. The next stage is the computation of EMD in order to derive the IMFs of sequency flux. For the computation of EMD the algorithm of [31] is used with the default settings provided by the authors.

The periodicities of sequency flux along with the periodicities of each IMF are estimated using a windowed-autocorrelation function. The (normalized) autocorrelation function is computed using a set of window lengths ranging from 2 s to 12 s with 1 s increments and a hop-size of 5 ms. The dominant periodicities of each frame per window length are estimated by finding the global maxima of the ACFs within the range of 30 to 600 BPM. Another set of periodicities is computed using the peaks of the ACFs that occur after the first minima, which in some cases may coincide with the global maxima.

A first set of tempo hypotheses for each IMF and the original sequency flux, and for each window length, is formed using the median values of the dominant periodicities (although it is acknowledged that the mode in some cases could give a more reliable estimate). Hypotheses that do not agree with the medians of the second set of periodicities are removed from this set. This "filtering" stage is done to ensure that the estimations are relatively stable, because if the ACF is noisy it may not exhibit a clear peak structure.

A second set of tempo hypotheses is formed using the medians of the second set of detected periodicities. Only tempi that are lower than the minimum tempo of the first set of hypotheses are kept. This "filtering" stage is specifically done to utilize the periodicities found in higher IMFs (i.e., lower tempi): in these IMFs the ACF decays slowly to zero and therefore, the peak of the ACF that occurs after the first minimum is a more robust estimator of the dominant tempo than the global maximum.

4.2 Induction

Depending on the complexity of the signal, some of the hypotheses formed according to the description given above might be wrong, including hypotheses derived from the original sequency flux. This could be due to various reasons such as: signals having a weak periodic structure and the inability of the ACF to detect periodicities within these structures; inaccurate IMFs generated from EMD due to over-iteration [28]; or even the choice of the summary statistic that is used to estimate the global tempo, which in this case is the median.

In the induction stage, we seek rational related tempi. For each hypothesis and from its respective waveform (i.e., sequency flux, or an IMF), we compute the total number of occurrences of the IOI that matches that hypothesis. The IOIs are rounded to the nearest BPM value of the ACF's resolution. This computation considers all possible interval pairs, not just consecutive ones. In the next step, we check whether integer multiples of the hypothesis that has the maximum number of IOI occurrences exist. If that is the case, that hypothesis along with its multiples is considered as valid.

If no multiples are found, we use the hypotheses derived

from sequency flux. As a reminder, there are as many hypotheses as the number of different window lengths, which are used in the computation of windowed ACF. From these hypotheses we choose the one that has the fastest tempo. Multiples and submultiples of this hypothesis (which may also be found in IMFs) are also considered to be valid. The decision to use the fastest tempo estimated from sequency flux, is based on the rationale that one has to first estimate the tatum before estimating the beat, or similarly, to first estimate the beat before estimating the measure.

4.3 Results

The proposed method was evaluated on twenty music excerpts provided by the 2006 Music Information Retrieval Evaluation eXchange campaign (MIREX06) for the Audio Tempo Extraction Competition from 2006 to 2018. The excerpts have a 30-seconds duration and are of constant tempo. The dataset includes a wide range of tempi, musical styles, genres, instrumentation and tempo estimations (derived from tapping-data) made by 40 listeners. It's also worth noting that some excerpts contain high levels of background noise, making them useful for testing algorithmic robustness in such scenarios. The ground-truth data consist of two tempi with their respective salience, because not all listeners were tapping on the same metrical level (e.g., the second tempo could be two, or three times faster than the first one). The algorithms were evaluated according to their ability to track both tempi correctly, and a *P*-score which relates the two estimations with their respective salience levels.

In this work, we are interested in evaluating: the accuracy of multiple tempo hypotheses formed in subsection 4.1; and whether the induction method presented in subsection 4.2 actually filters out the weak hypotheses while retaining the correct tempo. We also used a 4% tolerance level (*Acc1* value) on the estimations instead of the 8% used in the competition. The evaluation showed that the proposed method generates hypotheses that include at least one of the two tempi in 100% of the cases, and hypotheses that include both tempi in 85% of the cases. The induction algorithm preserved 95% of the correct tempo hypotheses by incorrectly rejecting one correct tempo, which indicates that the induction step could be improved.

Although these excerpts were meant to be used as a "training" dataset for the algorithms submitted to the MIREX campaigns, in this work this dataset was used as a case study. This allowed us to: (i) test whether there is any particular advantage of using a set of different window lengths when computing the ACF instead of just one; (ii) inspect the relative contribution of each IMF and sequency flux to the formation of correct tempo hypotheses; and (iii) identify pitfalls related to the induction process. Fig. 1 shows the number of occurrences of sequency flux and each IMF in the set of correct tempo hypotheses per excerpt. Notably, for some excerpts, the correct tempo was estimated exclusively from the IMFs and not from sequency flux. This is also the case for the slow tempi of the dataset, for which the correct tempo could only be estimated from the high IMFs.

5. CONCLUSIONS

We presented a multiresolution method that is capable of estimating multiple tempi from a detection function. In this work, the detection function was computed using the Walsh-Hadamard transform and it was therefore termed *sequency flux*, as opposed to the widely used spectral flux. The method acts on different time scales of the signal because it detects periodicities using a set of different window-lengths for the computation of the windowed-ACF, and because the EMD decomposes the sequency flux into signals that exhibit different characteristic time scales. In this way, the dominant periodicities of sequency flux that correspond to different time scales are emphasized and most importantly, these periodicities can be directly and more clearly observed from the IOIs of each IMF. The method is simple to implement and does not rely on empirically set thresholds or values that depend on the complexity of the signal (e.g., genre, instrumentation).

In general, the method generates some tempo hypotheses that could be considered as incorrect, and which are not harmonically related. This is the major reason why an induction stage was employed. However, the existence of metrical levels that are not harmonically related may indicate harmonic changes or rhythmic structures that occur *within* a "harmonic" time scale, but further work is needed to verify this hypothesis. In addition to the generation of non-harmonically related tempo hypotheses, in some cases, the method failed to estimate both tempi correctly, which indicates that some tempi cannot be directly derived from sequency flux or the IMFs. These results could be due to various reasons, some of which are already mentioned in Section 4. Some reasons that are specifically related to the EMD include the possible over-iteration, which may have occurred in the sifting stage, and the so-called "mode mixing" problem. Sifting the signal too much may have generated artificial modes that do not actually appear in the data [28]. The mode mixing problem refers to the presence of similar time scales in different modes, which may have distorted the median periodicity values found from these IMFs. This issue could potentially be addressed by using variants of the EMD such as the EEMD [32] or the CEEMD [33]. However, in this work the EMD was preferred over other methods because of its simplicity.

Clearly, a more sophisticated and robust induction algorithm is needed for this method to work on musical pieces with polyrhythms. However, the simple induction algorithm presented here achieves 95% accurate results on this case study but obviously, the method needs to be evaluated on larger datasets before making any claims related to its overall accuracy. Additionally, this method is currently not suitable for real-time applications due to its computational complexity. While the Fast Walsh-Hadamard transform is computationally more efficient than the Fast Fourier Transform (since it utilizes only real number additions and subtractions) the proposed method is still slow for real-time applications. This is because the ACF computation with multiple window lengths, the chosen EMD algorithm, and the current induction process, all contribute to the overall complexity. Nonetheless, the method finds

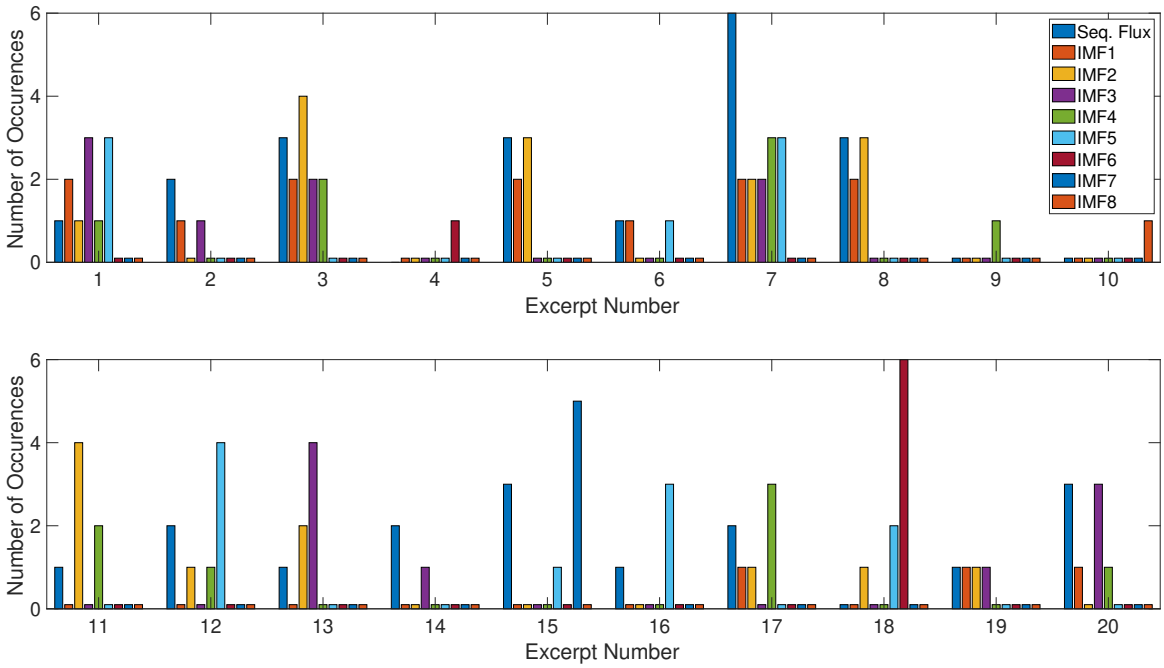


Figure 1. Number of occurrences of each IMF and sequency flux in the set of correct tempo hypotheses per excerpt.

many (offline) applications in areas such as: computational musicology (e.g., studying the interaction between rhythmic components in music); audio-to-MIDI transcription systems (e.g., capturing multiple rhythmic layers in polyphonic music); content-based music retrieval (e.g., searching based on rhythmic similarities); and automatic accompaniment generation (e.g., creating accompaniments that better follow a melody's rhythmic structure).

In the future, we plan to adapt this method for pieces with non-constant tempi and explore its beat tracking capability on different metrical levels. In conclusion, according to the results of this case study, our initial hypothesis is justified: the IMFs do carry tempo information on multiple hierarchical metrical levels. Hopefully, this information will be used to improve existing approaches related to meter induction in general, and more specifically to compound meter induction.

Acknowledgments

This research was co-financed by the European Regional Development Fund of the European Union and Greek national funds through the Operational Program Competitiveness, Entrepreneurship and Innovation, under the call Research - Create - Innovate (MIS 5149205).

6. REFERENCES

[1] H. Honing, "Without it no music: Beat induction as a fundamental musical trait," in *Annals of the New York Academy of Sciences*, 2012, pp. 85–91.

[2] H. Schreiber, J. Urbano, and M. Müller, "Music tempo estimation: Are we done yet?" in *Transactions of the Int. Society for Music Information Retrieval*, 2020.

[3] J. A. Bilmes, "Techniques to foster drum machine expressivity," in *Proc. Int. Computer Music Conference*, 1993.

[4] G. Kokkonis, N. Ordoulidis, G. Evagelou, and S. Kazazis, "Analysing tempo stability in Greek rebetiko music. the case of Vasilis Tsitsanis's repertoire," in *Digital Technologies Applied to Music Research. Methodologies, Projects and Challenges*, 2024.

[5] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," in *Journal of New Music Research*, 2002.

[6] O. Nieto and J. P. Bello, "Systematic exploration of computational music structure research," in *Proc. Int. Society for Music Information Retrieval Conference*, 2016.

[7] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proc. Int. Society for Music Information Retrieval Conference*, 2012.

[8] J. P. Bello, C. Duxbury, M. E. Davies, and M. B. Sandler, "On the use of phase and energy for musical onset detection in the complex domain," in *IEEE Signal Processing Letters*, vol. 11, 2004, pp. 553–556.

[9] A. Klapuri, A. J. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," in *IEEE Trans-*

actions on Audio, Speech, and Language Processing, vol. 14, 2006, pp. 342–355.

[10] P. Grosche and M. Müller, "Computing predominant local periodicity information in music recordings," in *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 33–36.

[11] S. Dixon, "Onset detection revisited," in *Proc. Int. Conference on Digital Audio Effects*, 2006.

[12] S. Böck and G. Widmer, "Local group delay based vibrato and tremolo suppression for onset detection," in *Proc. Int. Society for Music Information Retrieval Conference*, 2013.

[13] G. Percival and G. Tzanetakis, "Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses," in *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, 2014, pp. 1765–1776.

[14] S. Dixon, E. Pampalk, and G. Widmer, "Classification of dance music by periodicity patterns," in *Proc. Int. Society for Music Information Retrieval Conference*, 2003.

[15] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals." in *Journal of the Acoustical Society of America*, vol. 103, 1998, pp. 588–601.

[16] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 421–424.

[17] G. Peeters, "Template-based estimation of time-varying tempo," in *Advances in Signal Processing*, 2007, pp. 1–14.

[18] F. Gouyon, P. Herrera, and P. Cano, "Pulse-dependent analyses of percussive music," in *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, vol. 4, 2002, pp. IV–4174–IV–4174.

[19] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," in *Journal of New Music Research*, vol. 30, 2001, pp. 39 – 58.

[20] D. P. W. Ellis, "Beat tracking by dynamic programming," in *Journal of New Music Research*, vol. 36, 2007, pp. 51 – 60.

[21] O. Lartillot and D. Grandjean, "Tempo and metrical analysis by tracking multiple metrical levels using autocorrelation," in *Applied Sciences*, 2019.

[22] S. Böck, F. Krebs, and G. Widmer, "Accurate tempo estimation based on recurrent neural networks and resonating comb filters," in *Proc. Int. Society for Music Information Retrieval Conference*, 2015.

[23] H. Schreiber and M. Müller, "A single-step approach to musical tempo estimation using a convolutional neural network," in *Proc.Int. Society for Music Information Retrieval Conference*, 2018.

[24] S. Böck and M. E. P. Davies, "Deconstruct, analyse, reconstruct: How to improve tempo, beat, and downbeat estimation," in *Proc. Int. Society for Music Information Retrieval Conference*, 2020.

[25] A. Spanias, "A hybrid transform method for analysis/synthesis of speech," in *Proc. IEEE Global Telecommunications Conference GLOBECOM '91: Countdown to the New Millennium. Conference Record*, vol. 2, 1991, pp. 719–724.

[26] D. S. Stoffer, "Walsh-fourier analysis and its statistical applications," in *Journal of the American Statistical Association*, vol. 86, 1991, pp. 461–479.

[27] N. E. Huang, Z. Shen, S. R. Long, M. C. Wu, H. H. Shih, Q. Zheng, N.-C. Yen, C.-C. Tung, and H. H. Liu, "The empirical mode decomposition and the hilbert spectrum for nonlinear and non-stationary time series analysis," in *Journal of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, vol. 454, 1998, pp. 903 – 995.

[28] G. Wang, X. Chen, F.-L. Qiao, Z. Wu, and N. Huang, "On intrinsic mode function." in *Advances in Adaptive Data Analysis*, vol. 2, 2010, pp. 277–293.

[29] A. Pikrakis and S. Theodoridis, "An application of empirical mode decomposition on tempo induction from music recordings," in *Proc. Int. Society for Music Information Retrieval Conference*, 2007.

[30] K. Trohidis and L. J. Hadjileontiadis, "Tempo induction from music recordings using ensemble empirical mode decomposition analysis," in *Computer Music Journal*, vol. 35, 2011, pp. 83–97.

[31] G. Rilling, P. Flandrin, and P. Gonçalves, "On empirical mode decomposition and its algorithms," in *In Proc. of IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, 2003.

[32] Z. Wu and N. E. Huang, "Ensemble empirical mode decomposition: a noise-assisted data analysis method," in *Advances in data science and adaptive analysis*, vol. 1, 2009, pp. 1–41.

[33] M. E. Torres, M. A. Colominas, G. Schlotthauer, and P. Flandrin, "A complete ensemble empirical mode decomposition with adaptive noise," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, 2011, pp. 4144–4147.

CORRELATIONS BETWEEN OBJECTIVE AND SUBJECTIVE EVALUATIONS OF MUSIC SOURCE SEPARATION

Erika RUMBOLD (erumbold@uvic.ca) [1], **George TZANETAKIS** (gtzan@uvic.ca) [1], and **Bryan PARDO** (pardo@northwestern.edu) [2]

[1] *Department of Computer Science*, **University of Victoria**, BC Canada
[2] *Interactive Audio Lab*, **Northwestern University**, IL United States

ABSTRACT

The standard quality metric for audio source separation is the signal-to-distortion ratio (SDR), despite correlating poorly to human perception. In this paper, we investigate the correlation between subjective listener evaluations and multiple objective quality metrics, including the SDR. Music tracks are separated using five source separation models, and the resulting separation quality is calculated using five objective evaluation metrics. A listener study was conducted to also obtain subjective evaluations of the music separation quality. It was found that none of the observed objective metrics correlated well to human perception, with the best correlation coefficient being 0.246. We also found that the objective metrics did not agree with each other regarding which is the best and worst performing music source separation models.

1. INTRODUCTION

Music source separation is the task of isolating instruments, or groups of instruments, from a mixture of multiple instruments playing at once. For example, the group, or stem, for vocals would include background vocals in addition to the lead singer. The target stems can be specified in many ways; the MUSDB18 dataset for music separation [1] features the stems `bass`, `drums`, `vocals`, and `other`. Given a mixture of these four stems, the goal is to generate four waveforms that correspond to each of the original stems. In addition to being a well-studied topic on its own, source separation is used in a variety of other music information retrieval (MIR) applications, including automatic music transcription, lyric recognition, and music enhancement.

Source separation is continually advancing, improving on prior work. As with many scientific endeavors, this advancement relies on an evaluation step. For music source separation, this evaluation typically measures how "clean" the separated audio is (i.e., whether or not there are extra elements present that wouldn't be in a recording of the instrument by itself), and how complete it is (i.e., whether there are missing elements in the separated signal). There are many methods of evaluating audio quality, which fall under two categories: human evaluation, and automated methods. The latter can be further distinguished as either a closed-form formula that calculates an amount of error between the separator's output signal and the expected signal, or a statistical model that estimates a "goodness" score with information learned about either a data distribution [2] or human evaluation data [3].

With audio data being the focus of MIR tasks, the general goal should be to produce something that sounds good to human listeners. Human evaluation is the most direct way to determine whether we think some audio sounds good, but it is expensive and requires a lot of time to collect sufficient data to inform algorithm development. In response to these shortcomings, several automated error calculation methods have been proposed and used in various MIR tasks, including music source separation.

The most commonly used metric for audio source separation is the signal-to-distortion ratio (SDR), an error calculation that finds the ratio of unwanted sound (i.e., distortion) that occurs in an audio signal to the entirety of a target signal [4]. SDR is frequently cited in existing audio source separation work and serves as the baseline measure for many source separation challenges (e.g., Sony Demixing Challenge [5] and the MUSDB18 benchmark [1]).

Despite its popularity, SDR has been proven to correlate poorly to human perception [6, 7], meaning an audio signal that a listener would deem as having poor quality may still get a good SDR value, or vice versa. With the objective of making audio that sounds good to listeners, the correlation between evaluation metrics and human perception should be prioritized. This discrepancy has been acknowledged in the speech domain, prompting the development of new evaluation methods that achieve better correlation to human perception [8–17]; but this has not yet been realized to the same degree in the music domain.

In this work, we investigate existing evaluation methods for music source separation to determine if any of them achieve a strong enough correlation to human perception to be a reliable alternative to subjective human evaluation.

2. RELATED WORK

2.1 Subjective Evaluation of Audio Quality

Subjective evaluation of audio quality refers to the rating of audio stimuli by human subjects. Participants of a sub-

Copyright: © 2024 Erika Rumbold et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

jective listening study may be asked to rate the audio generally (i.e., "How good does the audio sound?"), or they could be asked to evaluate a specific attribute (e.g., level of interference, intelligibility, or musical intonation). In addition to the types of questions that can be asked, there are several evaluation protocols that are commonly used in audio research. Some are designed for the evaluation of individual stimuli one by one, while others have participants evaluate multiple stimuli in comparison with each other. These protocols also differ in the granularity of data received. Evaluations on a scale of 1 to 100, as in a MUSHRA study [18], tend to have tighter differences between responses than evaluations on a scale of 1 to 5.

In this paper, we conduct a Mean Opinion Score (MOS) study. Participants in a MOS assessment are asked to rate stimuli individually on a rational number scale, usually as follows: 1-Bad, 2-Poor, 3-Fair, 4-Good, and 5-Excellent. MOS is one of the most common subjective assessment methods in audio applications and is ideal to get a more general idea of human perception.

2.1.1 Crowdsourcing

Studies to acquire subjective data traditionally have been conducted in a laboratory, in which researchers can ensure a controlled environment, such as the required environmental features of standard MUSHRA. However, efforts have been made to adapt subjective quality assessments to an online crowdsourced format [19–22].

Typically, lab-based audio tests require participants to complete assessments in rooms that meet specific acoustic qualities, using the same technology as all other participants (e.g. headsets, operating systems, etc.). Some protocols additionally restrict participants to those that meet certain criteria such as level of expertise in an audio-related field or not having been diagnosed with a hearing disorder [18] in order to eliminate the possibility of a participant's evaluation being affected by extraneous noise or a lack of understanding of the task. However, conducting tests in a lab costs a significant amount of both time and money; and subjective assessment trials that take place in a lab attract fewer participants, leading to results that are less statistically significant [21].

These drawbacks have led to the rise of subjective evaluations that take place online. Many efforts have been made to adapt subjective assessment protocols to an online, crowd-sourced environment [19–22], forgoing some of the strict participant eligibility criteria and environmental control in favor of acquiring a larger and more diverse set of results. Moving these assessments online is also more cost effective; and it takes less time to acquire data than it takes in a lab. Traditional MUSHRA trials, for example, can take several hours for each participant to complete, depending on the number of trials completed. Online evaluation tasks, on the other hand, are designed to be completed much more quickly. Lab-based assessments typically have a small number of people evaluate a lot of things, whereas online assessments are taken by many more people, but each usually completes only a few tasks.

A shortcoming of online assessments is that they cannot be directly monitored, making it possible for results to be affected by the listening environment, the equipment used, or the participants' integrity. These effects cannot be screened ahead of time, but there are a few methods that can be implemented to filter online study results. For example, the CrowdMOS platform [21] asks participants for the type of listening device they used during the study (e.g., headphones, laptop speakers). It is expected that a person listening on speakers would not be able to hear finer details of audio as acutely as listeners using headphones. Cartwright, et al. [22] also asked about listening device in their web-based MUSHRA assessment, as well as the quietness of the room in which the participant completed the study. Asking these questions allows the researchers to eliminate data from participants that do not fit their criteria (e.g., using headphones, being in a quiet environment).

An online assessment can also contain a hearing test to assess the participants' hearing capabilities. For example, the MUSHRA assessment from Cartwright, et al. [22] features two hearing tests which require listeners to report how many tones they hear in a sequence. This sequence always includes a tone pitched at 55 Hz and at 10 kHz tone with up to 6 other tones being between those pitches. It is expected that a listener completing the study in a noisy room or with an inadequate listening device would not be able to hear the 55 Hz or 10 kHz tone. Researchers can also hide anchor questions within the survey, as is already the practice in MUSHRA. The answer should be obvious, so researchers can easily identify participants that did not understand the directions or intentionally submitted inaccurate responses. If participants do not answer these anchor questions correctly, their data can be eliminated.

Despite the need to prune crowdsourced results, crowdsourced assessments can achieve comparable results to those of their lab-based equivalents while costing significantly less and being quicker to execute [19]. More time may be necessary to screen crowdsourced results, but this is usually done computationally and does not take a significant amount of time from the researchers like in-lab assessments do.

2.2 Objective Evaluation of Audio Quality

Objective evaluation of audio is achieved without human subject data. Therefore, objective evaluation metrics are more practical for researchers to use, being significantly quicker and cheaper to execute than a subjective evaluation study. There are many methods of evaluation that can be applied to music source separation, and they typically take one of two forms: closed form equations that compute an amount of error, or models that predict an audio quality rating. Objective evaluation methods can also be classified by whether or not they require a ground truth signal to which the source separation model's output can be compared; every closed form equation method requires a ground truth.

2.2.1 Closed Form Evaluation Methods

The Signal-to-distortion ratio (SDR) [4] is the current standard evaluation metric for music source separation. An

estimate of source $\hat{s}_i$ is assumed to be composed of four separate components:

$$\hat{s}_i = s_{target} + e_{interf} + e_{noise} + e_{artif}$$

where $\hat{s}_i$ is the true source, and e_{interf}, e_{noise}, and e_{artif} are terms for interference, noise, and artifacts, respectively. These equations also assign equal weights to the different error terms. So it is assumed that each type of distortion contributes equally to the overall quality of the source $\hat{s}_i$ [6].

From these attributes, we are able to compute different energy ratios by the relation of these terms to the true source. SDR, or the overall measure of how good the source estimate sounds in comparison to the true source can be represented as:

$$\text{SDR} = 10\log_{10}\left(\frac{||s_{target}||^2}{||e_{interf} + e_{noise} + e_{artif}||^2}\right) \quad (1)$$

SDR is measured in decibels (dB), and higher values are better.

Since the original proposal of SDR, several issues with the metric have been discovered, including an easy way to boost one's scores by changing the amplitude scaling of source estimates. This prompted the proposal of a version of SDR that is not dependent on amplitude scaling, SI-SDR [23]. They first rescale the target s by finding the orthogonal projection of the estimate $\hat{s}$ on the line spanned by s. The scaled reference is denoted as e_{target}, which allows us to break down the estimate $\hat{s}_i$ as $\hat{s}_i = e_{target} + e_{res}$. From this, we can define SI-SDR by the equation

$$\text{SI-SDR} = 10\log_{10}\left(\frac{||e_{target}||^2}{||e_{res}||^2}\right) \quad (2)$$

As with SDR, SI-SDR is measured in decibels (dB), and higher values are better. And despite the potential improvements SI-SDR has over SDR, SDR remains the standard evaluation metric for the task of music source separation.

Although SDR is the established standard, most loss functions that are normally used in neural network training can also be used to evaluate the quality of audio source separation. For example, L1 and L2 losses can be used to evaluate the similarity between an estimated signal and the target signal. These loss functions have been previously implemented in music source separation, being parts of the training architectures for source separation algorithms Demucs [24, 25] and Spleeter [26]. L1 can be observed as the absolute error, and L2 as the squared error.

In the context of music source separation, these calculations are typically done on the power spectrograms of the target and estimate signals, and lower values are better.

2.2.2 *Audio Quality Prediction Models*

The second type of objective evaluation is an audio quality predictor, or in other words, a non-human system (i.e., neural network) that is trained using existing audio evaluation data to predict the evaluation of other audio stimuli. These evaluation methods can be further distinguished by the type of data on which they are trained. The PEASS Toolkit[1] [10] and MOSNet [14] are two evaluation systems that are trained on human data - MUSHRA and MOS, respectively - to predict quality scores for the input audio. Alternatively, audio quality predictors can be trained on data that is another quality evaluation model's output. For example, Quality-Net [11] is a speech quality assessment model that is trained on PESQ [3] data and outputs a PESQ score prediction for an audio input. PESQ, or Perceptual Evaluation of Speech Quality, is a model developed for telephone networks and codecs that predicts Mean Opinion Score.

A shortcoming of audio quality prediction models operating in the music domain that is being addressed in the speech domain is the requirement of an available target signal, or ground truth separated signal. The previously mentioned speech models, Quality-Net and MOSNet, are two examples of evaluators that only take as training inputs estimated signals and their PESQ or MOS scores, respectively. PEASS, however, requires the target signal of each stem and the estimated mixture signal as inputs. This is a significant issue when no target audio is available.

An alternative approach to a "referenceless" model is given by the Fréchet Audio Distance (FAD) [2]. Inspired by the Fréchet Inception Distance (FID) [27], which was developed to evaluate generative models for images, FAD compares statistics computed on a set of estimated signals to reference statistics computed on a large set of studio recorded music.

FAD uses a VGGish [28] model to generate embeddings for the reference set and the evaluation set. Like how Fréchet Audio Distance, it is derived from Fréchet Inception Distance, VGGish is derived from the VGG image recognition architecture. Multivariate Gaussians are computed on both the evaluation set embeddings $N_e(\mu_e, \Sigma_e)$ and the reference embeddings $N_r(\mu_r, \Sigma_r)$; and Dowson, et al. [29] define the Fréchet distance between two Gaussians as:

$$F(N_b, N_e) = ||\mu_b - \mu_e||^2 + tr(\Sigma_b + \Sigma_e - 2\sqrt{\Sigma_b \Sigma_e}) \quad (3)$$

where tr is the trace of a matrix. As a distance measure, lower FAD scores are better. FAD was developed for the task of music enhancement, but the metric could also be effective for music source separation.

3. LISTENER STUDY

To make comparisons between objective and subjective evaluation methods, we performed source separation on a subset of MUSDB18 using five selected algorithms. We then conducted a crowdsourced listener study to collect MOS ratings, and measured the separation quality according to five objective metrics. These evaluations and the correlation between them are discussed further in Section 4.

3.1 Dataset

MUSDB18 consists of 150 stereo mixtures of songs, about 10 hours of data, that span a variety of genres. The song files are encoded at 44.1kHz and in the Native Instruments stems format. This multitrack format is composed of five

[1] PEASS can operate in both the music and speech domains.

stereo streams corresponding to the mixture, drums, bass, other, and vocals. MUSDB18 is among the most-cited datasets for existing work in music source separation.

From the 150 songs in the MUSDB18 dataset, we curated a subset of 30 songs, mostly from the test set, that represented a wide range of musical genres and were balanced between male and female singers. We did not randomly select songs because MUSDB18 skews in favor of male singers and the Pop/Rock genre; manual curation makes it easier to represent the genres and vocals that are less common in the MUSDB18 data set as a whole.

Each of the 30 MUSDB18 songs was truncated to a 7-second segment. This duration is short enough to be separated efficiently while also being long enough for listeners to effectively evaluate. These segments were screened to ensure that the clip contained enough of each stem - bass, drums, and vocals - to be evaluated.

The source separation algorithms used in this experiment are referred to as **HDX** [25], **DMX** [24], **D3N** [30], **SPL** [26], and **SSW** [31]. These were chosen due to their rankings on the Papers with Code leaderboard for average SDR, seeking algorithms that represented the top, middle, and bottom tiers. In addition to comparing the correlations of existing metrics to human perception, we can determine the reliability of an SDR-based leaderboard when considering human perception.

Each of these separators outputs tracks corresponding to the stems `bass`, `drums`, `vocals`, and `other`. We disregarded the other track in this experiment due to its ambiguity. The `other` stem could contain a keyboard synthesizer or a harp - a solo saxophone or an entire string orchestra.

Ignoring the `other` track leaves us with 30 tracks separated by five source separating systems into three stems, or a data set of 450 stems to evaluate.

3.2 Listening Assessment

We conducted a MOS study in which participants were asked to rate two separate attributes of the audio that was presented: the level of "other instruments" present, and the level of "artifacts" present. To clarify the term for participants without audio training, "artifacts" were defined in the introduction as "extra sound that cannot be recognized as a musical instrument or voice."

3.2.1 Participants

Participants were recruited and paid through Amazon Mechanical Turk (MTurk), a platform for crowdsourcing user studies. They were paid $2.00 for each 10-question study they completed. Participants were required to be at least 18 years of age, which is enforced by MTurk, and they were strongly encouraged to complete the study with headphones or earbuds in a quiet environment.

3.2.2 Pre-Screening

The subjective assessment started with a hearing screening similar to the screening defined by Cartwright, et al. [22] in their online MUSHRA assessment. Participants were first asked to adjust the volume of a 1000 Hz sine wave to a comfortable level and encouraged to not change the level afterward.

They then listened to two 8-second audio clips and counted how many separate sine wave tones they heard. Each clip contained at least a 55 Hz and a 10 kHz tone, with the possibility of up to six more tones between 55 Hz and 10 kHz. It is expected that a participant in a suitable listening environment with an appropriate listening device should be able to hear the 55 Hz and 10kHz tones.

Participants had three attempts to answer both screening questions correctly. Incorrect answers would be followed by a prompt for the participant to change their listening environment or device and try again. Failing this check three times would prompt the participant to submit their responses; they would not be able to view the rest of the study and they would not be compensated.

3.2.3 Procedure

Following the hearing screening, a description of the rating system was given to the participants who passed the hearing test. It was explained that they would give two ratings for each audio example on a 1-5 scale - one for presence of artifacts and another for presence of sounds from other instruments. A rating of 1 indicated "Bad: a lot of sound from other instruments or artifacts", and 5 indicated "Excellent: no sound from other instruments or artifacts." They were presented with example audio clips and descriptions of what was meant by presence of other instruments and presence of artifacts.

Each assessment consisted of 10 audio clips of the same stem type - bass, drums, or vocals, and no audio clip was repeated across the assessment versions. The assessments were released in batches grouped by stem type; so one batch would only contain audio clips of drums, for example. A participant could decide to complete each assessment in the batch, or just a few. MTurk does not have the capability to randomize the order in which assessments appear in a batch; so to ensure the latter assessments were taken enough times, an assessment in the batch was unpublished when it had been completed by 10 participants. Assessments that were submitted with a failed hearing test were republished until it had been completed by 10 participants who passed the hearing test.

For each of the 10 questions on an assessment, participants were asked to listen to a 7-second audio clip in its entirety, then separately rate the level of "other instruments" and "artifacts" present in the clip. At the end of the assessment, participants were asked to report the type of listening device they used to complete the assessment and a rating on a 1-5 scale of how quiet their listening environment was throughout the assessment, where 5 meant no noise and 1 meant extremely noisy. They were provided spaces to report any changes to their listening environment that may have occurred, as well as any additional comments.

4. RESULTS & DISCUSSION

In this section, we review the responses received from the listener study described in the previous section, and com-

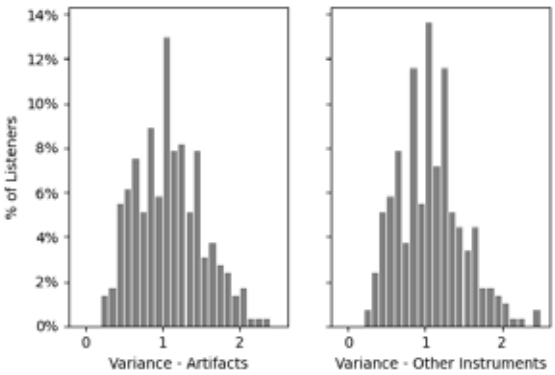


Figure 1. Distribution of variance between subjective ratings and the MOS of each audio example.

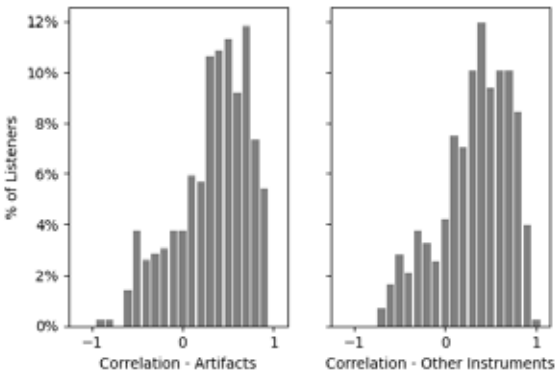


Figure 2. Distribution of Spearman rank correlation coefficients between subjective ratings and the MOS of each audio example.

pare those results to the evaluation of selected objective metrics.

4.1 Listener Responses

Fig. 1 shows the distribution of variance between an individual rating of an audio example and the MOS of that example. For both types of distortion, artifacts and other instruments, most individual rater scores varied, on average, by about 1.0 from the MOS. This amount of variation seems logical since participants can only respond with integers within the small range of 1 through 5; and it would be unlikely for a listener to rate an audio example as a 5 when the majority rate it as a 1.

We also investigated the Spearman rank correlation between the ratings a listener gave and the mean opinion scores for the same songs. The distribution of these correlation coefficients is shown in Fig. 2. Most listeners had a correlation coefficient around 0.5 to the MOS of the examples they rated. It also shows a significant number of assessments that were negatively correlated to the others. Specifically, 78 out of 450 assessments had negatively correlated ratings; 25 bass assessments, 28 drum assessments, and 25 vocals assessments.

	SDR	SI-SDR	L1	L2	FAD
Bass	-0.301	-0.235	0.238	0.246	0.076
Drums	0.003	-0.021	-0.006	-0.027	-0.020
Vocals	-0.006	-0.055	0.035	0.023	0.204

Table 1. Spearman rank correlation coefficients for each objective metric compared to MOS for presence of artifacts

	SDR	SI-SDR	L1	L2	FAD
Bass	-0.088	-0.043	-0.026	-0.000	-0.122
Drums	0.017	0.027	0.034	0.023	-0.036
Vocals	0.027	0.000	0.049	0.053	-0.003

Table 2. Spearman rank correlation coefficients for each objective metric compared to MOS for presence of other instruments

We considered excluding data that had a negative correlation. However, it is possible that these participants genuinely heard the audio differently. Given the information acquired through the listening assessment, it would be impossible to prove that these participants, or which of them, were not completing the study with integrity.

We also noticed that many submissions from unique participants had either all 4s and 5s as their ratings, or all 1s and 2s. Because the audio clips were created with source separation algorithms of varying quality, according to average SDR, responses like these are highly unlikely. We investigated the correlation on a subset of the responses that excluded those that gave all of the same or two adjacent ratings. However, excluding such data did not significantly affect the distributions of variance or correlation. Given these observations, the rest of this paper uses the full set of subjective evaluation responses.

4.2 Correlation Between Objective and Subjective Evaluation

We have chosen five existing objective metrics to examine: SDR [4], scale-invariant SDR (SI-SDR) [23], L1 loss, L2 loss, and Fréchet Audio Distance (FAD) [2]. SDR serves as the baseline for these experiments, being the current standard metric for music source separation. SI-SDR was chosen to see if the scale-invariant aspect affects the outcome. L1 and L2 losses are typically used in training prediction models, but not in the final evaluation of audio quality. FAD is a music evaluation metric that does not require the ground-truth target signal.

First, we compare the MOS of each audio example against the corresponding score from each objective metric, as shown in Fig. 3 and 4. Clearly, there isn't a strong correlation in any of the plots, regardless of metric or stem type.

To confirm this notion, we can look at the Spearman's rank correlation coefficients, shown in Table 1 and Table 2. We can see that the strongest positive correlation occurred with L1 and L2 loss for the MOS of artifacts present in bass examples. With a correlation coefficient of 0.246, however, this is still not a strong relationship. This implies that, as MOS improves, only a quarter of L1 and L2 evaluations do

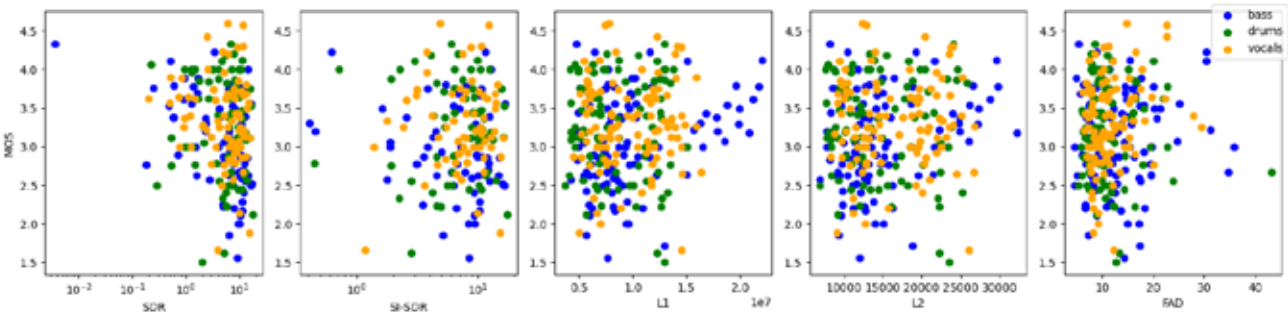


Figure 3. Objective metrics vs. MOS for presence of artifacts. The axes for SDR and SI-SDR are log-scaled.

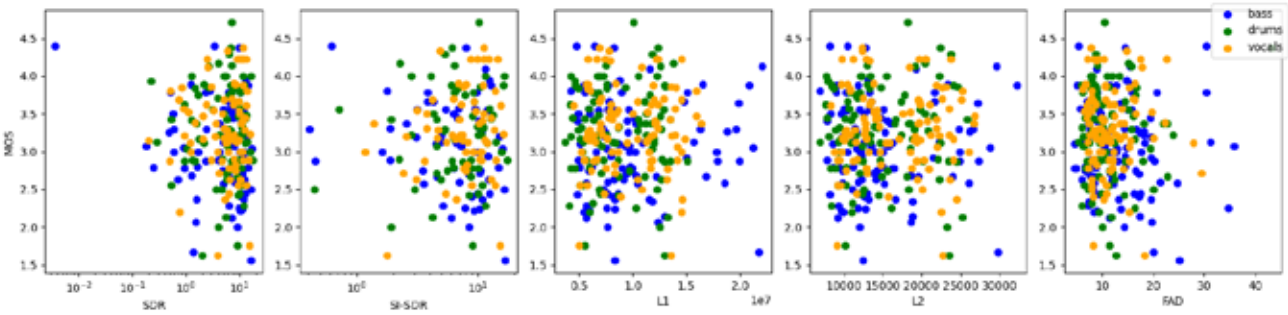


Figure 4. Objective metrics vs. MOS for presence of other instruments. The axes for SDR and SI-SDR are log-scaled.

so as well.

4.3 Ranking Source Separation Performance by Different Evaluation Criteria

In addition to correlation analysis, we considered how each evaluation metric would rank the performance of the five source separation algorithms. We also compare the ranking from the observed metrics to that from the MUSDB18 leaderboard [2] .

4.3.1 Rank Order from Objective Metrics

As shown in Table 3, no observed metric maintained the same exact rank order as the MUSDB18 leaderboard. This includes SDR, which the leaderboard uses. The difference between the average SDR from this experiment and that from the leaderboard is likely due to using a subset of the MUSDB18 instead of the full 50 song test set.

Among the five objective metrics, the order most similar to the MUSDB18 leaderboard was given by L2 loss, placing **HDX** as the best separator and **SPL** and **SSW** as the worst. The least similar rankings to the leaderboard were from FAD, which only agreed with **SSW** being at the bottom.

4.3.2 Rank Order from Listener Study

We also looked at the rank orders from the objective metrics in comparison to the collected mean opinion scores, also shown in Table 3. We can see that the rank order according to MOS is completely different from the order according to both the MUSDB18 leaderboard and the observed objective evaluation metrics.

These subjective results may tell us that the outputs of these source separation models are more similar than SDR and other objective metrics would indicate. However, there are other factors that could have affected the listeners' responses. Despite being provided with examples of what to listen for, participants may not have fully understood the task. They could have also been affected by the environment in which they completed the study. The responses at the end of the assessment showed that 27.55% of the surveys were completed in an environment that was "somewhat noisy" or worse. There is also the possibility that participants who reported an appropriate level of environmental noise could not have been genuine. These are both caveats that come with conducting an online crowdsourced study.

5. CONCLUSION

We have investigated the relationship between objective and subjective evaluation of music source separation. We collected mean opinion score data for separated stems from five different source separation algorithms, and compared those results to evaluations from five different objective metrics.

With the goal of producing audio that sounds good to listeners, the objective metrics used in MIR work would ideally correlate to human evaluation. We found that none of the observed objective metrics correlated well with the listener opinion data, with the best correlation being 0.246. We also found that no evaluation metric agreed on which source separation algorithm was the best, worst, or in be-

[2] https://paperswithcode.com/sota/music-source-separation-on-musdb18

Rank	MUSDB18	SDR	SI-SDR	L1	L2	FAD	MOS (Artif.)	MOS (Inst.)
(Best) 1	HDX	HDX	HDX	D3N	HDX	D3N	SSW	SSW
2	DMX	DMX	DMX	HDX	D3N	SPL	D3N	SPL
3	D3N	SPL	SPL	DMX	DMX	HDX	SPL	D3N
4	SPL	SSW	SSW	SPL	SPL	DMX	HDX	HDX
(Worst) 5	SSW	D3N	D3N	SSW	SSW	SSW	DMX	DMX

Table 3. Relative rank order of source separation algorithms according to the MUSDB18 leaderboard, the five observed objective metrics, and the mean opinion score (MOS) when listening for presence of "Artifacts" or "Other Instruments."

tween. On the MUSDB18 leaderboard, there was a 4.1 dB difference in average SDR between the best and worst separator used in this paper. However, human listeners evaluated all five separators very similarly and in a vastly different order.

5.1 Future Work

We confirm that the most commonly used evaluation metric for music source separation does not achieve a strong correlation to the opinions of listeners, and none of the other observed metrics does either. One could use these insights to develop a new evaluation metric with the intent of correlating well to human perception. While possible, a robust dataset of audio as well as listener evaluation data would be required to achieve this goal. It is also possible that listeners would be able to hear the differences in audio quality more acutely if the audio examples were presented in comparison with each other, instead of one at a time. If so, a MUSHRA assessment would be worth pursuing.

6. REFERENCES

[1] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "Musdb18-a corpus for music separation," 2017.

[2] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A metric for evaluating music enhancement algorithms," *arXiv preprint arXiv:1812.08466*, 2018.

[3] A. Rix, J. Beerends, M. Hollier, and A. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No.01CH37221)*, vol. 2, 2001, pp. 749–752 vol.2.

[4] E. Vincent, R. Gribonval, and C. Fevotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[5] Y. Mitsufuji, G. Fabbro, S. Uhlich, and F.-R. Stöter, "Music demixing challenge 2021," *arXiv preprint arXiv:2108.13559*, 2021.

[6] E. Cano, D. FitzGerald, and K. Brandenburg, "Evaluation of quality of sound source separation algorithms: Human perception vs quantitative metrics," in *2016 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 1758–1762.

[7] D. Ward, H. Wierstorf, R. D. Mason, E. M. Grais, and M. D. Plumbley, "Bss eval or peass? predicting the perception of singing-voice separation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 596–600.

[8] X. Dong and D. S. Williamson, "A classification-aided framework for non-intrusive speech quality assessment," in *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019, pp. 100–104.

[9] ——, "An attention enhanced multi-task model for objective speech assessment in real-world environments," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 911–915.

[10] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–2057, 2011.

[11] S. Fu, Y. Tsao, H. Hwang, and H. Wang, "Quality-net: An end-to-end non-intrusive speech quality assessment model based on BLSTM," *CoRR*, vol. abs/1808.05344, 2018. [Online]. Available: http://arxiv.org/abs/1808.05344

[12] S.-W. Fu, C.-F. Liao, Y. Tsao, and S.-D. Lin, "Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2031–2041.

[13] S.-W. Fu, C. Yu, T.-A. Hsieh, P. Plantinga, M. Ravanelli, X. Lu, and Y. Tsao, "Metricgan+: An improved version of metricgan for speech enhancement," *arXiv preprint arXiv:2104.03538*, 2021.

[14] C. Lo, S. Fu, W. Huang, X. Wang, J. Yamagishi, Y. Tsao, and H. Wang, "Mosnet: Deep learning based objective assessment for voice conversion," *CoRR*, vol. abs/1904.08352, 2019. [Online]. Available: http://arxiv.org/abs/1904.08352

[15] P. Manocha, A. Finkelstein, R. Zhang, N. J. Bryan, G. J. Mysore, and Z. Jin, "A differentiable perceptual

audio metric learned from just noticeable differences," 2020.

[16] P. Manocha, Z. Jin, R. Zhang, and A. Finkelstein, "Cdpam: Contrastive learning for perceptual audio similarity," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 196–200.

[17] Z. Zhang, P. Vyas, X. Dong, and D. S. Williamson, "An end-to-end non-intrusive model for subjective and objective real-world speech assessment using a multi-task framework," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 316–320.

[18] B. Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[19] K.-T. Chen, C.-C. Wu, Y.-C. Chang, and C.-L. Lei, "A crowdsourceable qoe evaluation framework for multimedia content," in *Proceedings of the 17th ACM international conference on Multimedia*, 2009, pp. 491–500.

[20] M. Cartwright, B. Pardo, and G. J. Mysore, "Crowdsourced pairwise-comparison for source separation evaluation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 606–610.

[21] F. Ribeiro, D. Florêncio, C. Zhang, and M. Seltzer, "Crowdmos: An approach for crowdsourcing mean opinion score studies," in *2011 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2011, pp. 2416–2419.

[22] M. Cartwright, B. Pardo, G. J. Mysore, and M. Hoffman, "Fast and easy crowdsourced perceptual audio evaluation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 619–623.

[23] J. L. Roux, S. Wisdom, H. Erdogan, and J. R. Hershey, "Sdr – half-baked or well done?" in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 626–630.

[24] A. Défossez, N. Usunier, L. Bottou, and F. Bach, "Music source separation in the waveform domain," *arXiv preprint arXiv:1911.13254*, 2019.

[25] A. Défossez, "Hybrid spectrogram and waveform source separation," *arXiv preprint arXiv:2111.03600*, 2021.

[26] R. Hennequin, A. Khlif, F. Voituret, and M. Moussallam, "Spleeter: a fast and efficient music source separation tool with pre-trained models," *Journal of Open Source Software*, vol. 5, no. 50, p. 2154, 2020.

[27] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[28] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "Cnn architectures for large-scale audio classification," in *2017 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2017, pp. 131–135.

[29] D. Dowson and B. Landau, "The fréchet distance between multivariate normal distributions," *Journal of multivariate analysis*, vol. 12, no. 3, pp. 450–455, 1982.

[30] N. Takahashi and Y. Mitsufuji, "D3net: Densely connected multidilated densenet for music source separation," *arXiv preprint arXiv:2010.01733*, 2020.

[31] F. Lluís, J. Pons, and X. Serra, "End-to-end music source separation: is it possible in the waveform domain?" *arXiv preprint arXiv:1810.12187*, 2018.

RECONSTRUCTING THE CHARLIE PARKER OMNIBOOK USING AN AUDIO-TO-SCORE AUTOMATIC TRANSCRIPTION PIPELINE

Xavier RILEY (j.x.riley@qmul.ac.uk) and **Simon DIXON** (s.e.dixon@qmul.ac.uk) (0000-0002-6098-481X)

Centre for Digital Music, **Queen Mary University London**, UK

ABSTRACT

The Charlie Parker Omnibook is a cornerstone of jazz music education, described by pianist Ethan Iverson as "the most important jazz education text ever published". In this work we propose a new transcription pipeline and explore the extent to which state of the art music technology is able to reconstruct these scores directly from the audio without human intervention. Our pipeline includes: a newly trained source separation model for saxophone, a new MIDI transcription model for solo saxophone and an adaptation of an existing MIDI-to-score method for monophonic instruments.

To assess this pipeline we also provide an enhanced dataset of Charlie Parker transcriptions as score-audio pairs with accurate MIDI alignments and downbeat annotations. This represents a challenging new benchmark for automatic audio-to-score transcription that we hope will advance research into areas beyond transcribing audio-to-MIDI alone.

Together, these form another step towards producing scores that musicians can use directly, without the need for onerous corrections or revisions. To facilitate future research, all model checkpoints and data are made available to download along with code for the transcription pipeline. Improvements in our modular pipeline could one day make the automatic transcription of complex jazz solos a routine possibility, thereby enriching the resources available for music education and preservation.

1. INTRODUCTION

The automatic transcription of music from audio to a readable score represents a challenging area of research within the field of music information retrieval (MIR). Success in this task has the potential to revolutionise the way musicians learn, teach, and preserve music. Despite its significance, the task of converting an audio recording to a score of similar quality to those of professional engravers remains a challenging and open research area.

Among the many genres of music, jazz stands out as particularly complex due to its improvisation, syncopation, and swing-feel elements that are quintessential yet difficult to quantify and transcribe. The work of Charlie Parker, a leading figure in the development of bebop and modern jazz, demonstrates many of these characteristics. Parker's solos are not only technically demanding but also helped to define styles and idioms that have been adopted by subsequent generations of jazz musicians. This helps to explain why the Charlie Parker Omnibook [1] has been in print continuously since its publication in 1978. This collection of 60 transcriptions of his solos is a widely used resource for jazz students and professionals alike.

This paper investigates a pipeline approach to automatic music transcription (AMT) that leverages recent advances in source separation, MIDI transcription, and score generation technologies. By focusing on the solos of Charlie Parker, we aim not only to contribute a new tool to the field of music education but also to address a gap in the literature concerning the automatic transcription of complex jazz solos from audio to score. The choice of Charlie Parker as the subject of this study is deliberate: his work presents a challenging benchmark for evaluating the efficacy of AMT systems on material that is culturally significant and still widely studied today.

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

2. RELATED WORK

The transcription pipeline we propose in this work incorporates several topics, each of which could be discussed in depth. Due to length constraints we give an overview of important work and refer to more detailed summaries elsewhere where necessary.

2.1 Jazz transcriptions

Transcription has been an essential part of jazz education and musicology as jazz is mainly an aural tradition with few written sources. Regarding Charlie Parker specifically, an early example of academic work was Thomas Owens's (1974) PhD thesis [2], in which he transcribed a large number of Parker's solos leading to the identification of approximately one hundred melodic formulas. These range from concise four-note motifs to expansive multi-measure phrases. Even today, this extensive analysis provides inspiration for what automated approaches may enable.

Shortly afterward, the Charlie Parker Omnibook was published in 1978 and has been in print ever since. This contains 60 solos transcribed by trumpeter Ken Sloane with assistance from saxophonist Jamey Aebersold.

In the age of computational analysis, digital formats and scores are required. The most ambitious work to date is the

Weimar Jazz Database [3] [1] which encompasses approximately 200,000 tone events from 456 monophonic solos by 78 jazz masters, providing high-quality score and MIDI transcriptions. These transcriptions were carried out with a semi-automated pipeline involving manual corrections. The dataset includes 12 works by Charlie Parker.

Interest in generative modelling of Parker's improvisations led to the release of a digitised version of the Omnibook [4]. 50 of the 60 scores were released in MusicXML format to aid further research.

More recently, the FiloSax dataset by Foster and Dixon [5] includes 240 full length scores of jazz saxophone performances along with audio stems and copious metadata. These resources are key to enabling the transcription pipeline which we describe later in this work.

2.2 Source separation

Music source separation has seen a huge increase in interest with the advent of capable neural network models. The objective is to extract an individual part (stem) from an audio mixture by learning a source specific filter. This is a large and active research area and we refer elsewhere [6] for a recent survey and discussion of state-of-the-art approaches.

For this work we use Demucs [7], a widely used solution proposed by researchers at Facebook AI. We note that separation into drums/bass/vocals/other is widely available due to the MDB18 dataset [8], however to date there are no freely available models for separation of saxophone. This work addresses this gap by training and releasing a solo saxophone separation model which separates into "sax" and "other" stems. Details of the training procedure are discussed in section 3.2.

2.3 Transcription to MIDI

Automatic music transcription generally aims to convert musical audio into a useful symbolic representation such as MIDI. Benetos et al. [9] claim that monophonic transcription is a solved problem, however recent work [10] suggests that this is not true for more intricate sources like jazz saxophone and traditional Irish flute.

Historically, approaches to the monophonic transcription task have typically involved framewise pitch estimation followed by some method to segment the pitch contour into discrete notes. Pyin [11] (in particular the Pyin notes function) is still a popular tool for this purpose, but its accuracy when classifying at the note level is not sufficient to avoid manual corrections. CREPE Notes [10] improves on this but it is still limited to 82.31% note level F-measure on the FiloSax dataset.

Higher levels of accuracy have been achieved on piano recordings in recent years due to high quality datasets and several capable machine learning architectures for transcription [12]. The issue of instruments beyond the piano was recently addressed [13, 14] by using scores and existing transcriptions which were realigned to audio to produce training data.

[1] https://jazzomat.hfm-weimar.de/dbformat/dboverview.html

2.4 Score layout

The process of converting a human performance to a musical score is typically performed by highly skilled human transcribers. Computational approaches have had limited success so far in this task. Performance timing must be interpreted in terms of musical units on a metrical grid and note durations must be chosen which allow the score to be parsed with minimal difficulty by human performers.

The qparse library [15] handles monophonic score layout by combining these two stages, using a probabilistic grammar defined by the user. We adopt this method in our work, training a new grammar based on Charlie Parker's notated scores, as described in section 3.2.

Quantising a note to its nearest position on the grid is an obvious approach, however this does not always account for rhythmic variation in expressive human performance such as the use of swing feel and playing "behind the beat". We explore these issues further in section 5. Polyphonic score layout is more challenging and beyond the scope of this work and we refer the reader to [16] for more information.

2.5 Other pipelines and alternative approaches

The separate-transcribe-notate pipeline that we propose has been used before, however not in a fully automated setting. Work on transcribing jazz bass [17] used a similar approach but the result required a number of manual corrections.

Large scale jazz transcription has similarly relied on automated approaches with corrections by human annotators [3,18]. The Dig that Lick project [19] attempted fully automatic extraction of MIDI data for jazz solos but did not include the conversion to musical score. Their approach [20] extracted main melody contours directly from audio mixtures which were then segmented into notes. Melodia [21] was the previous state of the art for melody estimation from monophonic and polyphonic sources. An alternative approach [22, 23] combines the source separation and transcription stages as a joint learning problem, but these techniques do not produce score output.

In the latter stages of completing this work, we became aware of Martínez et al. [24], who present an approach for end-to-end audio-to-score (A2S) transcription, specifically applied to a newly constructed corpus of real and synthetic saxophone recordings. The corpus is smaller than FiloSax (3 hours vs. over 20 hours) and contains simpler melodic material such as musical scales and short phrases. Their method attempted to predict a text-based score representation (Kern) directly as model output, which differs from the pipeline approach we have taken in this work. They obtained similar results for their synthetic and real world audio data which suggests that the approach is viable, however the model weights were not available at the time of writing so we were unable to perform a direct comparison. This will be a subject of future work.

In our work, we chose to investigate a more modular pipeline, so that any improvements in performance of system components will directly benefit our pipeline.

3. METHOD

3.1 Dataset production

We used an existing corpus of digital scores from the Omnibook [4]. This consists of 50 of the 60 original tracks in the published Omnibook, digitised to MusicXML. We hope to digitise the remaining 10 tracks as part of future work. We sourced recordings that corresponded to each transcribed segment and uploaded these to the SoundSlice platform [2]. The first author then added downbeats manually to all 50 tracks, using the variable speed and fine adjustment settings that the platform offers. Only sections corresponding to the notated scores were annotated with downbeats. This typically includes the main theme and the saxophone solo. In most cases these form a continuous segment but additional metadata is included to account for gaps where other solos occur.

Given the age of the source material (recorded in live and studio settings before Parker's death in 1955) there were a number of issues with performances not being reproduced at concert pitch (A440Hz). We first attempted to correct for this using an automatic approach [3] however on review these were found to be inaccurate, potentially due to the poor recording quality. We chose to use the SoundSlice pitch adjustment feature to make manual global pitch adjustments. These adjustments were applied to the source audio files using the rubberband library [4].

With downbeats and intonation resolved, we then aligned the human transcriptions (in MIDI representation) to the audio of the solo saxophone stem. This was done using Dynamic Time Warping (DTW, as implemented in the SyncToolbox [25]). To fine-tune the DTW path, the MIDI note onsets were aligned to the frame level activations of a transcription model using a high-resolution alignment procedure [14].

To aid further research, we make the dataset freely available to other researchers. This includes MusicXML scores, downbeats, performance-aligned MIDI files, links to the original recordings and extracted saxophone stems tuned to A440Hz standard. These can be accessed via the supplementary site [5].

3.2 Transcription Pipeline

We now describe the proposed transcription pipeline for the task of transcribing a score from an audio recording containing saxophone.

The first stage is beat estimation, for which we chose Madmom [26] for its strong performance and convenient implementation. This estimates the beat, downbeat and time signature from the original audio mix which are later used in score estimation. Initially we found that the default parameters led to a number of failures on the Omnibook recordings. To address this we first obtain a rough estimate of global tempo from the transcribed MIDI using the distribution of inter-onset intervals [27]. We also constrain the minimum tempo to 15 BPM below the prior estimate (to avoid picking a tempo which is too slow) and the maximum to 350 BPM (to account for the extreme tempi found in bebop music). Empirically this was found to improve tempo estimation for this dataset.

The next stage is to separate the saxophone from the audio mixture. The Demucs source separation model (version 3) [7] was trained using audio from the FiloSax dataset [5]. The complete set of 240 tracks was used for training to maximise the amount of available training data, with tracks divided into 6 second segments. The model was trained for 180 epochs with a learning rate of 0.001 and a batch size of 8.

For transcribing the separated saxophone stem to MIDI, we train a new model based on the high resolution piano transcription work by Kong et al. [12]. We take an 80-10-10% split (by pieces) of the FiloSax dataset. Each recording is divided into 10-second samples, with a hop size of 1 second. We used a learning rate of 10^{-4} and a batch size of 8. The training data came from tenor saxophone and so audio was augmented with random pitch shifts, continuously distributed between ± 5 semitones. This helps to match the registers of the alto and baritone saxophones respectively and increases training variety given the relatively small dataset. Training ran for 27.5K steps, approximately 55 epochs, scaling the learning rate by 0.9 every 5K steps.

Finally for the MIDI-to-score stage we used the "qparse" package [15]. This handles rhythmic quantisation of MIDI elements along with score engraving by treating it as a parsing problem using a user-specified grammar which defines probabilistic transition states between note types. We first defined a grammar based on the transitions observed in the Omnibook data using the method in [28], however this led to an overly complex grammar which led us to simplify some of the rules empirically. This was done to reduce the frequency of triplets in favour of quavers in the predicted scores. The final grammar definition is available with the code that is released alongside this work via the supplementary site.

4. RESULTS

4.1 Beat tracking

We include results for automatically estimated downbeats in Table 1. The method used is described in section 3.2. From the results we see that the automatic estimation is generally poor, albeit with a wide variance. As all the pieces in the dataset are in 4/4, we also include results for the optimal downbeat placement within a bar (i.e. selecting the best of all possible "rotations"). The improvement when rotations are included suggests that the beat tracking is relatively accurate, but that downbeat estimation is challenging on this source material. Since it requires little user effort to specify the correct rotation, it may be desirable to compute multiple transcriptions for each possible downbeat rotation and allow the end-user to select the best

[2] https://www.soundslice.com
[3] https://librosa.org/doc/main/generated/librosa.estimate_tuning.html#librosa.estimate_tuning
[4] https://breakfastquay.com/rubberband/
[5] https://aim-qmul.github.io/SaxTranscriptionPipeline/

Method (Madmom)	Mean F-measure	Max
w/ init bpm	29.20 ± 25.11	92.52
w/ init bpm and rotations	61.27 ± 30.57	97.74

Table 1. Downbeat F-measure results calculated by *mir_eval* for the proposed Charlie Parker dataset. The values are expressed as percentages and a default tolerance of 70ms was used.

Method	SDR (dB)
LALAL.AI Wind Stem	14.03 ± 2.38
Ours (Demucs + FiloSax)	**14.22** ± 2.10

Table 2. Source Separation: Signal-to-Distortion Ratio (SDR) results calculated by *mir_eval* for two methods. The SDR values are expressed in decibels (dB) and include the standard deviation. Higher is better.

result.

4.2 Solo saxophone source separation

We evaluate our source separation model on a private dataset of 10 professionally produced jazz quartet recordings featuring saxophone. To the best of our knowledge, there are no recent source separation models for saxophone discussed in the literature, therefore we have compared our results with a commercial offering [6] to provide a baseline. As we can see in Table 2, our method performs better, but it should be noted that the Lalal.ai results were for a more general "Wind" stem, whereas ours was specifically trained for saxophone.

4.3 MIDI transcription

This stage of the pipeline was evaluated on two datasets using standard metrics for note-level accuracy (no offsets, 50ms tolerance) provided by the mir_eval library. Evaluation results for a test split of the FiloSax dataset are shown in Table 3, and results for the Charlie Parker Omnibook dataset are shown in Table 4.

Here we estimate a MIDI transcription using the source separated saxophone stems as input, which is then compared with the aligned MIDI created from the ground truth scores. We evaluate our method alongside CREPE Notes, which previously demonstrated state-of-the-art results on FiloSax over a number of other methods [10]. We also include results for Başaran et al. [20], as this was the method selected to produce automated transcriptions for the Dig That Lick project.

On the FiloSax data, our method demonstrates a large improvement over previous methods, but this should be taken with caution as the test data is from the same distribution as the training set. On the out-of-distribution Omnibook data, our method also achieves the highest F-measure score, but the results are considerably lower than those achieved for FiloSax. The three systems tested gave results that were respectively over 12%, 16% and 20% worse than the Filosax

[6] https://www.lalal.ai/

	P_{50}	R_{50}	F_{50}
CREPE Notes [10]	82.35	83.05	82.67
Başaran et al. [20]	87.68	92.57	90.01
Ours	**96.47**	**95.90**	**96.19**

Table 3. Note level transcription (FiloSax): Results for note-level transcription accuracy (no offsets) on a 25 track test split of FiloSax. P_{50}, R_{50}, and F_{50} are Onset-only Precision, Recall and F1-measure, expressed as percentages, at 50ms resolution.

	P_{50}	R_{50}	F_{50}
CREPE Notes [10]	70.93	70.63	70.41
Başaran et al. [20]	70.60	**77.98**	73.68
Ours	**74.70**	76.96	**75.43**

Table 4. Note level transcription (Omnibook): Results for note-level transcription accuracy (no offsets) on the proposed Omnibook dataset alignments using stems from section 4.2. Abbreviations are described in Table 3.

results, suggesting that the Omnibook can be considered a very challenging dataset.

4.4 Score timing

The task of producing a score from a MIDI representation of a human performance has received less attention than the audio-to-MIDI stage and as a consequence there is less agreement in the literature on which metrics are most appropriate. We have used a subset of the metrics proposed by Cogliati et al. [29] to give an indication of the edit distance between our transcribed scores and the ground truth.

For comparison we include a baseline from the Music21 package [30], which is used to import the transcribed MIDI with `quantizePost=True`. This is to give an indication of score quality when handled by basic quantisation alone.

The results in Table 5 show mean error rates, normalised by the number of notes in each piece. We find that our proposed qparse workflow requires fewer edits than the basic quantisation method, showing particular efficiency regarding the number of rests that need to be edited (34.50% insertions vs 791.93% for the baseline method). This means our method requires less work to reach the level of a human transcribed score, however there is still room for improvement.

We also present some excerpts of transcribed scores with their associated ground truth in Figure 1 and Figure 2. The complete set of scores is available for detailed comparison via the supplementary site [7]. From these we see that our proposed method performs well at producing "readable" scores, at the expense of some rhythmic accuracy. Offbeat eighth notes are often quantised to be on the beat, for example, which suggests that further work is required to refine the grammar rules provided to qparse. In Figure 2 we

[7] https://aim-qmul.github.io/SaxTranscriptionPipeline/

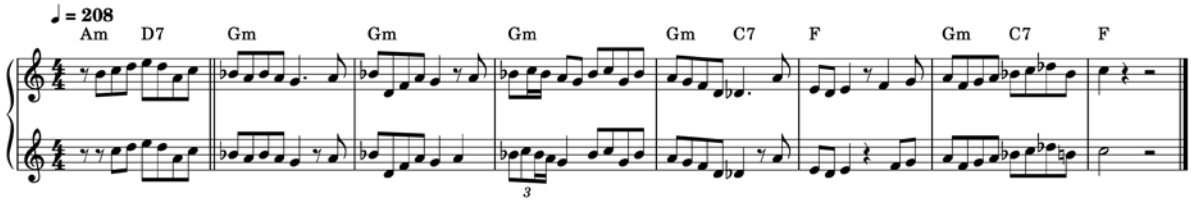


Figure 1. An extract of "Marmaduke" with ground truth on the upper staff and our transcription on the lower staff.

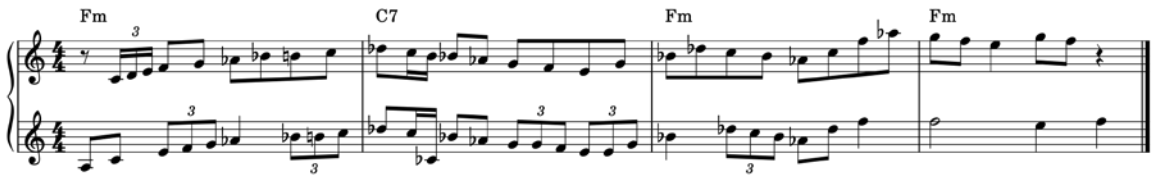


Figure 2. An extract of "The Bird" with ground truth on the upper staff and our transcription on the lower staff.

	Note		Rest		
	Insert	Del	Insert	Del	TimeSig
qparse [15]	23.26	54.74	34.50	80.23	2.55
baseline [30]	45.36	48.41	791.93	41.26	2.69

Table 5. Score quantization: Mean error rates for score timing expressed as percentages of the ground truth score length. Results are shown for the 31 scores that qparse could successfully process. The baseline is described in section 4.4.

also see issues with tracking pitches in the upper register of the alto, as discussed in Section 5 below.

4.4.1 Stability of qparse

In this work we adopted the latest version of qparse, however we encountered a number of issues that remain unresolved at the time of writing. This resulted in only 31 out of 50 scores being available for comparison, with the others failing to find a valid parsing result. In the released code we provide a fallback option to offer more basic parsing in cases where qparse fails, and we will continue to work with qparse authors to improve the robustness of the code.

5. DISCUSSION

In attempting to reconstruct this corpus of saxophone transcriptions, we have encountered several challenges described in the prior sections. In spite of these, we are pleased to note that our proposed pipeline obtains the highest accuracy of existing methods for the audio-to-MIDI stages and that the MIDI-to-score stage outputs useful scores in many cases.

Some of the difficulties we faced in producing scores from performance MIDI were surprising, even in the relatively simple monophonic case which our work focused on. We hope to explore this issue further in upcoming work to find a robust method for conversion from performance timing to score timing. The goal would be to handle the trade off between rhythmic accuracy and score complexity. We also consider that renewed research into metrics and measures of score transcription quality would be beneficial in this area.

A particularly challenging problem was that of beat detection, which traditionally assumes that the performers share a concept of a single shared pulse. In jazz however, there has been some work to suggest that participatory discrepancies (PDs) are a central feature of the music. This is where a soloist, for example, might "lay back" and phrase their part deliberately late [31]. This idea is obliquely supported in a 1949 DownBeat interview with Charlie Parker himself: when pushed to define the new genre of "bop" he gave the answer:

> "The beat in a bop band is with the music, against it, behind it," Charlie said. "It pushes it. It helps it."

This suggests that his placement of notes ("the music") was deliberately manipulated in reference to the pulse ("the beat") and indeed, this is something we observe in our data. For example, we see that short note values at the ends of bars are often quantised to the start of the next bar. Given the lack of synchrony between performers, it is clear that future MIDI-to-score solutions need to be flexible with the concept of a downbeat in order to succeed in transcribing jazz in a human-like fashion.

The separation and MIDI transcription parts of our pipeline are shown to perform well, however we feel that their generalisation to all types of saxophone could be enhanced further. Empirically, our system struggles with the extreme upper register of the alto saxophone (altissimo, harmonics etc.) but we believe this is due to the available

training data being in a conservative range of the tenor saxophone. This could be improved in future iterations with access to more diverse data (including alto saxophone) and with more aggressive data augmentation strategies.

Despite these challenges, we do find that the best case outputs of our proposed method produce usable scores. With further refinement of this pipeline we take a step towards transcribing music accurately at scale. This opens up new possibilities for management of large collections, and also the potential for time savings for the transcription workflows of individual musicians.

6. CONCLUSIONS

In this work we present a new pipeline for end-to-end transcription of solo saxophone from audio to score. Each stage in the pipeline is discussed and evaluated where possible, with state-of-the-art results for source separation of saxophone and MIDI transcription of solo saxophone.

Alongside the transcription pipeline, we release a new dataset of audio-score pairs of 50 Charlie Parker recordings. These feature accurate alignments and timing information which allow for use in tasks such as transcription, expressive performance analysis and generative modelling.

We also publish our automated transcriptions alongside the ground truth scores to demonstrate what is currently possible in difficult cases of monophonic transcription from audio to score. The results are not yet at the level of a human expert, but are a promising step towards more reliable transcription methods in future. We hope these will serve as a baseline to stimulate future research in this area.

Acknowledgments

The first author is a research student at the UKRI Centre for Doctoral Training in Artificial Intelligence and Music, supported by UK Research and Innovation [grant number EP/S022694/1]. We would like to thank Florent Jacquemard and Lydia Rodriguez de la Nava for their cooperation and assistance with using qparse for this work.

7. REFERENCES

[1] C. Parker, *Charlie Parker Omnibook*, J. Aebersold and K. Slone, Eds. Hal Leonard, 1978.

[2] T. Owens, "Charlie Parker: Techniques of improvisation," Ph.D. dissertation, University of California, Los Angeles, 1974.

[3] M. Pfleiderer, K. Frieler, J. Abeßer, W.-G. Zaddach, and B. Burkhart, *Inside the Jazzomat: New Perspectives for Jazz Research*. Schott Music GmbH, 2017.

[4] K. Déguernel, E. Vincent, and G. Assayag, "Using multidimensional sequences for improvisation in the OMax paradigm," in *13th Sound and Music Computing Conference*, 2016.

[5] D. Foster and S. Dixon, "Filosax: A dataset of annotated jazz saxophone recordings," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

[6] S. Uhlich, G. Fabbro, M. Hirano, S. Takahashi, G. Wichern, J. L. Roux, D. Chakraborty, S. Mohanty, K. Li, Y. Luo *et al.*, "The sound demixing challenge 2023," *arXiv preprint arXiv:2308.06981*, 2023. [Online]. Available: https://arxiv.org/abs/2308.06979

[7] A. Défossez, "Hybrid spectrogram and waveform source separation," in *Proceedings of the ISMIR 2021 Workshop on Music Source Separation*, 2021.

[8] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "MedleyDB: A multitrack dataset for annotation-intensive MIR research," in *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, 2014, pp. 155–160.

[9] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2019.

[10] X. Riley and S. Dixon, "CREPE Notes: A new method for segmenting pitch contours into discrete notes," in *Proceedings of the 20th Sound and Music Computing Conference*, Stockholm, Sweden, 2023, pp. 1–5.

[11] M. Mauch and S. Dixon, "PYIN: A fundamental frequency estimator using probabilistic threshold distributions," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 659–663.

[12] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang, "High-resolution piano transcription with pedals by regressing onset and offset times," *IEEE ACM Transactions on Audio, Speech and Language Processing*, vol. 29, pp. 3707–3717, 2021.

[13] B. Maman and A. H. Bermano, "Unaligned supervision for automatic music transcription in the wild," in *International Conference on Machine Learning ICML*, ser. Proceedings of Machine Learning Research, vol. 162, 2022, pp. 14 918–14 934.

[14] X. Riley, D. Edwards, and S. Dixon, "High resolution guitar transcription via domain adaptation," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024. [Online]. Available: https://qmro.qmul.ac.uk/xmlui/handle/123456789/94701

[15] F. Foscarin, F. Jacquemard, P. Rigaux, and M. Sakai, "A parse-based framework for coupled rhythm quantization and score structuring," in *Mathematics and Computation in Music: 7th International Conference, MCM 2019, Madrid, Spain, June 18–21, 2019, Proceedings 7*. Springer, 2019, pp. 248–260.

[16] L. Liu, Q. Kong, V. Morfi, and E. Benetos, "Performance MIDI-to-score conversion by neural beat tracking," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference (ISMIR)*, 2022, pp. 395–402.

[17] X. Riley and S. Dixon, "Filobass: A dataset and corpus based study of jazz basslines," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2023, pp. 500–507.

[18] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "GuitarSet: A dataset for guitar transcription," in *International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

[19] L. Henry, K. Frieler, G. Solis, M. Pfleiderer, S. Dixon, F. Höger, T. Weyde, and H.-C. Crayencour, "Dig that lick: Exploring patterns in jazz with computational methods," *Jazzforschung / Jazz Research*, vol. 50/51, 2024, to appear.

[20] D. Başaran, S. Essid, and G. Peeters, "Main melody estimation with source-filter NMF and CRNN," in *19th International Society for Music Information Retrieval Conference*, 2018, pp. 82–89.

[21] J. Salamon and E. Gómez, "Melody extraction from polyphonic music signals using pitch contour characteristics," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 6, pp. 1759–1770, 2012.

[22] K. W. Cheuk, K. Choi, Q. Kong, B. Li, M. Won, J. Wang, Y. Hung, and D. Herremans, "Jointist: Simultaneous improvement of multi-instrument transcription and music source separation via joint training," *CoRR*, vol. abs/2302.00286, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.00286

[23] E. Manilow, P. Seetharaman, and B. Pardo, "Simultaneous separation and transcription of mixtures with multiple polyphonic and percussive instruments," in *2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 771–775.

[24] J. C. Martínez-Sevilla, M. Alfaro-Contreras, J. J. Valero-Mas, and J. Calvo-Zaragoza, "Insights into end-to-end audio-to-score transcription with real recordings: A case study with saxophone works," in *INTERSPEECH 2023*, Aug. 2023, pp. 2793–2797.

[25] M. Muller, Y. Ozer, M. Krause, T. Pratzlich, and J. Driedger, "Sync toolbox: A python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software*, vol. 6, no. 64, p. 3434, 2021. [Online]. Available: https://doi.org/10.21105/joss.03434

[26] 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 2016 ACM Conference on Multimedia, MM 2016*. ACM, 2016, pp. 1174–1178.

[27] S. Dixon, "Automatic extraction of tempo and beat from expressive performances," *Journal of New Music Research*, vol. 30, no. 1, pp. 39–58, 2001.

[28] F. Foscarin, F. Jacquemard, and P. Rigaux, "Modeling and learning rhythm structure," in *Sound and Music Computing Conference (SMC)*, 2019.

[29] A. Cogliati and Z. Duan, "A metric for music notation transcription accuracy," in *Proceedings of the 18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 407–413.

[30] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," in *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010, pp. 637–642.

[31] M. Butterfield, "Participatory discrepancies and the perception of beats in jazz," *Music Perception*, vol. 27, no. 3, pp. 157–176, 2010.

EXPLORING DEEP LEARNING METHODOLOGIES FOR MUSIC EMOTION RECOGNITION

Pedro L. LOURO (pedrolouro@dei.uc.pt) (0000-0003-3201-6990) [1],
Hugo REDINHO (redinho@dei.uc.pt) (0009-0004-1547-2251) [1],
Ricardo MALHEIRO (rsmal@dei.uc.pt) (0000-0002-3010-2732) [1, 2],
Rui Pedro PAIVA (ruipedro@dei.uc.pt) (0000-0003-3215-3960) [1], and
Renato PANDA (panda@dei.uc.pt) (0000-0003-2539-5590) [1, 3]

[1] *CISUC, DEI*, **University of Coimbra**, Coimbra, LASI, Portugal
[2] *School of Technology and Management*, **Polytechnic Institute of Leiria**, Leiria, Portugal
[3] *Ci2 - Smart Cities Research Center*, **Polytechnic Institute of Tomar**, Tomar, Portugal

ABSTRACT

Classical machine learning techniques have dominated Music Emotion Recognition (MER). However, improvements have slowed down due to the complex and time-consuming task of handcrafting new emotionally relevant audio features. Deep Learning methods have recently gained popularity in the field because of their ability to automatically learn relevant features from spectral representations of songs, eliminating such necessity. Nonetheless, there are limitations, such as the need for large amounts of quality labeled data, a common problem in MER research. To understand the effectiveness of these techniques, a comparison study using various classical machine learning and deep learning methods was conducted. The results showed that using an ensemble of a Dense Neural Network and a Convolutional Neural Network architecture resulted in a state-of-the-art 80.20% F1-score, an improvement of around 5% considering the best baseline results, concluding that future research should take advantage of both paradigms, that is, conbining handcrafted features with feature learning.

1. INTRODUCTION

In the early stages of developing Music Emotion Recognition (MER) systems, the focus was mainly on classical machine learning (ML) techniques, which involved a significant amount of effort devoted to feature engineering [1, 2]. Music classification involves identifying gaps in dimensions such as melody, harmony, rhythm, dynamics, timbre, expressivity, texture, and form. Feature extraction algorithms capture these dimensions, and ML models are trained on them. However, current works mostly use low- and mid-level descriptors. Panda et al. [3] achieved 76% accuracy with a combination of novel emotionally relevant features based on audio analysis and a newly proposed dataset, the 4QAED dataset, surpassing the 69% accuracy plateau observed in the MIREX challenge held in 2007. The most pressing issue is the challenging and time-consuming process of designing such features, requiring expertise in signal processing, musicology, and ML to produce improvements.

As a way to avoid this process, deep learning (DL) has recently seen a rise in popularity due to its ability to automatically learn relevant features from raw input data. Recently, various DL methods have been applied to tackle MER, many of which employ convolutional (CNN) and recurrent (RNN) neural networks [4, 5]. Different approaches have been proposed for processing raw input data in audio applications. This includes end-to-end architectures [6, 7], transfer learning from larger datasets [8], and using embeddings extracted from pre-trained CNNs [9].

These techniques have limitations due to requiring large amounts of quality labeled data. Previous systems have applied audio transformations to increase the training set for chosen algorithms, but the impact of this approach for MER is not well known in comparison with other tasks such as genre recognition [10].

Neural networks lack interpretability due to their black-box nature, making it unclear which features are learned and extracted during training. In MIR, concerns have been raised about their ability to learn relevant information. However, studies such as the one by Choi et al. [11] showed that a 5-layer CNN can learn to extract features closely related to melody, harmony, percussion, and texture. In the same direction, Won et al. [12] demonstrated that a self-attention mechanism can learn relevant instrument, genre, and emotion detection information through heatmaps.

In this article, we have performed a comparative study of different classical ML and DL methodologies applied to MER. This study aims to understand these methods' effectiveness, considering the promising paths of DL-based approaches. To conduct this study, we have used the 4QAED dataset along with a recent expansion. We have explored various methodologies, including architectural improvements, audio augmentation techniques, alternative input data representations, and knowledge transfer from related tasks. Additionally, the expansion of the baseline dataset has allowed us to assess the impact of small dataset size

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

increases on the classification accuracy of DL models.

This study produced several contributions, including a state-of-the-art F1-score of 80.20% achieved through an ensemble of Dense Neural Network (DNN) and CNN architecture while considering data augmentation. Additionally, we conducted a comprehensive comparison of various methodological enhancements for solving MER, as well as an analysis of the influence of dataset size and class balancing on classification performance. To encourage reproducible research, we provide the code used for the conducted experiments [1].

2. BACKGROUND

The connection between music and emotions has been a focus of research in music psychology. Emotion from a musical piece can be examined through expressed, perceived, and induced perspectives. Perceived emotion provides the highest level of objectivity and is the focus of most works in the literature [13].

There have been several proposals to represent the range of human emotions. These can be divided into categorical models, which cluster similar emotions together, such as Hevner's Adjective Circle [14], and dimensional models, which create a multi-dimensional plane with axes representing different biological systems theorized to process emotion in the human brain. The most widely accepted dimensional model is Russell's Circumplex Model [15], seen in Figure 1, according to the literature. Scholars have raised concerns about categorical and dimensional models. Categorical models do not reflect the continuous nature of emotions, while dimensional models are complex and require prior knowledge for accuracy [16].

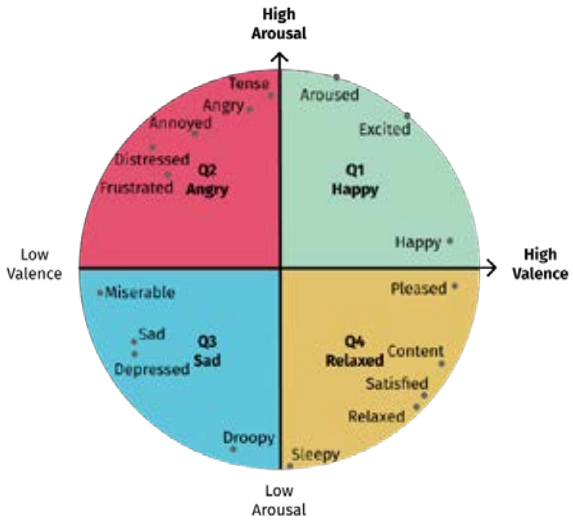


Figure 1. Russell's Circumplex Model. It is possible to represent emotions either as continuous values, which can be seen as individual points on a map, or as discrete labels that encompass a wider range of emotions.

In our team's research, Panda et al. [3] introduced the previously mentioned 4QAED dataset, which utilizes labels provided by experts from the AllMusic API. The labels were converted into A-V values [2], which correspond to the x- and y-axes of Russell's model, representing valence and arousal, respectively. The dataset takes a categorical approach by grouping all annotations into one of four quadrants rather than the continuous approach of the model. Please refer to the following section for a more detailed understanding of the dataset and its expansion.

3. METHODS

The methodologies explored for this work are presented in this section. We began by defining both ML and DL baseline methodologies in Section 3.1 and evaluating them on multiple datasets for comparison purposes.

The following section discusses the various methodologies that we explored and what motivated us to consider them. These approaches include improving the architecture by utilizing time-related information (Section 3.3), learning features from segments of entire samples (Section 3.4), obtaining alternative input representations through high-dimensional projections (Section 3.5), generating more training data through sample synthesis (Section 3.6), and utilizing learned information from related tasks (Section 3.7).

3.1 Baseline Architectures

We began our experiments by establishing a classic baseline using the state-of-the-art system by Panda et al, discussed previously, consisting on the top 100 standard and novel features, ranked using a feature selection algorithm, fed to a Support Vector Machine (SVM) classifier. For each dataset, we fine-tuned the hyperparameters of the SVM classifier using the same set of optimal features that were used in the original work.

Our team previously created a CNN architecture based on the research done by Choi et al. [17]. This architecture serves as the baseline for deep learning development. We modified the original architecture to process the extracted features on a small DNN that predicts one of the four quadrants from Russell's model, instead of outputting a binary vector. To prevent overfitting, an early stopping strategy was implemented, halting training when the training set accuracy reached a value equal to or greater than 90%, as found from previous experimentation. Unless explicitly stated otherwise, these points are the default configuration for the remaining approaches described in this section.

Regarding the optimal hyperparameters found for each methodology using a neural network as classifier, please refer to Table 1.

3.2 Explored Methodologies

We started off by examining the latest DL-based methodologies proposed for MER, focusing solely on systems that improve static emotion in music using only audio.

[1] https://github.com/g0ld3nl34f/exploring-dl-for-mer

[2] https://tivo.stoplight.io/docs/music-metadata-api/ZG9jOjQ3NjAxNTk-introduction

Table 1. Input sizes and best hyperparameters found for DL-based baseline and explored methodologies.

Methodology	Input		Best Hyperparameters			
	Type	Size	Epochs	BS	Optimizer	LR
DL Baseline	Mel-spectrogram	942×128 (≈30s)	200	150	SGD	1e-2
Baseline with GRU	Mel-spectrogram	942×128 (≈30s)	200	150	SGD	1e-2
CRNN	Mel-spectrogram	942×128 (≈30s)	200	50	SGD	1e-3
Hybrid	Mel-spectrogram and	942×128 (≈30s)	100	300	SGD	1e-2
Augmented	Handcrafted Features	1714	100	300	SGD	1e-2
ShortChunk CNN	Mel-spectrogram	116×128 (≈3.5s)	100	50	SGD	1e-3
Sample CNN	Waveform	59049 (≈3.5s)	150	50	SGD	1e-3
Baseline + TFM	Mel-spectrogram	942×128 (≈30s)	200	150	SGD	1e-2
Baseline + SB	Mel-spectrogram	942×128 (≈30s)	200	150	SGD	1e-2
Baseline + RG	Mel-spectrogram	942×128 (≈30s)	200	150	SGD	1e-2
Artists CNN	Mel-spectrogram	129×128* (≈3s)	200	100	Adam	1e-2
CRNN MTAT	Mel-spectrogram	942×128 (≈30s)	200	16	Custom**	Custom**

*A sample rate of 22.05kHz was used per the original implementation instead of 16kHz for the remaining methodologies.
**See Section 3.7 for details.

An important resource for our experiments is the work by Won et al. [18]. Here, a set of architectures from previous works on automatic music tagging are gathered and tested against each other. These comprise simple CNN-based architectures, end-to-end approaches, and even theory-motivated ones, all available in a GitHub repository [3] .

3.3 Architecture Improvements

We enhanced our DL baseline's architecture by incorporating two Gated Recurrent Units (GRU) [19] to enable learning time-domain-specific features. This was done as a starting point to improve the overall architecture. We also experimented with an implementation of the CRNN architecture, which was adapted from the previously mentioned work. This way it is possible to assess the DL baseline's feature learning portion ability to preserve information related with time.

A simple ensemble of a baseline CNN and a DNN, both previously pre-trained, performed remarkably well, referred as Hybrid Augmented form herein. The DNN portion was fed with all 1714 features found to be relevant in the same work used as a basis for the classical baseline. The fused information is then post-processed by a smaller DNN. This approach combined the information extracted from both paradigms to enhance the overall classification, with the addition of synthetic samples for pre-training the CNN portion. The complete architecture can be seen in Figure 2.

3.4 Segment-level Approaches

In our previous work, we utilized the complete 30-second samples accessible on 4QAED as the model's input. However, humans can easily recognize emotions in smaller samples. Considering the small size of our datasets, breaking down these samples into smaller segments can benefit us by increasing the number of training examples, which is an indirect way of data augmentation. A straightforward approach that follows this idea is introduced in [18] as ShortChunk CNN. During the model's training phase, each segment was treated as an individual sample, while for testing, all the predictions related to the segments of a sample were combined to obtain the final prediction, which is known as a many-to-one approach.

Previous deep learning works commonly used convolutional layers to extract features from spectral representations, which require specific parameters. However, the ideal parameters are not architecture-independent, requiring optimization should any of the layers change. To avoid this issue, Lee et al. [6] propose an end-to-end architecture, the Sample CNN. They suggest working directly with the raw audio signal and using a sequence of one-dimensional convolutional blocks, similar to the two-dimensional variant, and processing the output through a dense layer. It is important to note that these models were originally designed to output one of a set of labels, which varied depending on the dataset used. They were later translated from PyTorch to TensorFlow and reworked to output categorical labels.

3.5 Data Representations

As mentioned earlier, using Mel-spectrograms might not be the best approach for training a machine learning model to classify emotions. Embeddings, popular in Natural Language Processing (NLP) due to providing a smaller, more efficient representation of the location of words in sentences in a lower-dimensional space, are considered as an alternative to spectral representations. Recently, Koh et al. [9] applied this idea to audio by using the OpenL3 deep audio embedding library [4] and training classical ML techniques classifier on its output. The embeddings are derived from a Mel-spectrogram representation, resulting in a feature matrix of size 298×512.

The study's baseline dataset yielded a 72% F1-score with the Random Forest (RF) classifier from the scikit-learn library [5] , which is almost similar to the classical baseline. The experiment was further extended to the baseline

[3] https://github.com/minzwon/sota-music-tagging-models

[4] https://github.com/marl/openl3

[5] https://scikit-learn.org/stable/

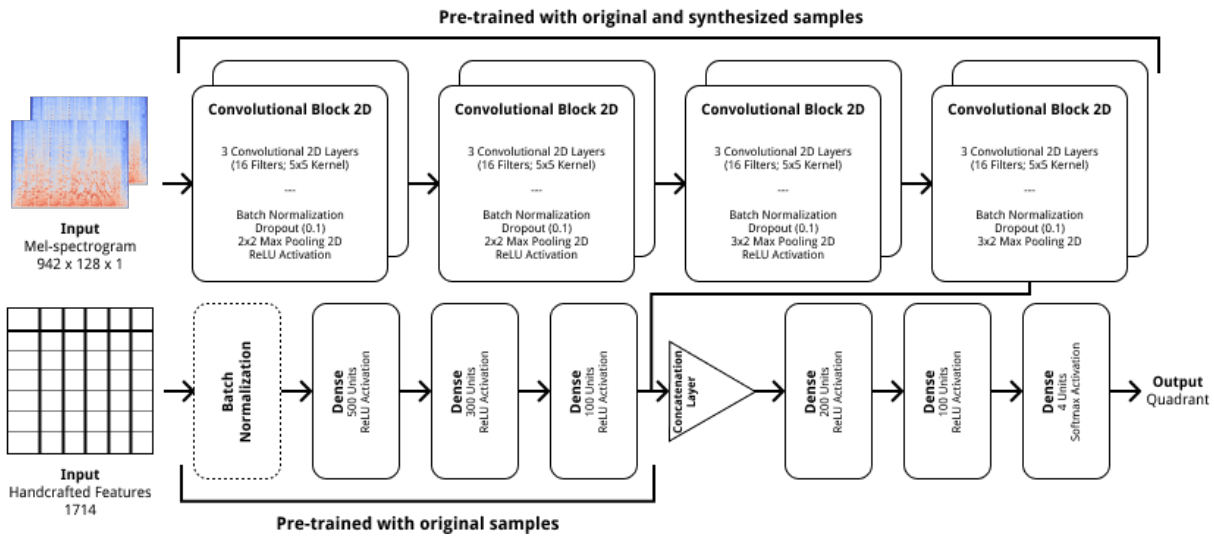


Figure 2. Hybrid Augmented architecture. Both the CNN and feature processing DNN are previously pre-trained on training data, with the addition of synthesized samples for the latter. Information from the aforementioned networks are concatenated and further processed by a smaller DNN.

dataset extension, and the embeddings generated by the autoencoder used in the DeepSMOTE approach, described in the next section, were also tested for comparison purposes.

3.6 Data Augmentation

We have delved deeper into both classical and deep learning (DL) approaches for augmenting data. In the classical approach, we have applied various audio augmentation techniques directly to the audio signal of a sample. These techniques randomly increase or decrease a factor associated with the transformation, such as time shifting (shifts start or end by five seconds), pitch shifting (increasing or decreasing pitch by two semitones), time stretching (speeding up or slowing down by 50%), and power shifting (increasing or decreasing amplitude by 10 dB), transformations used to obtain the synthesized sample for the Hybrid Augmented methodology. More of these techniques were experimented using the audiomentations library [6], including:

- Time-Frequency Masking (TFM), popular in the field of SER, which applies a mask over a portion of the time- and frequency-domain [20];
- Seven-Band Parametric Equalization (SB), applying a seven-filter pass on the sample, changing its timbre in the process;
- Random Gain (RG), randomly increasing or decreasing the loudness of a sample.

A factor is chosen randomly from a set of predefined intervals for each transformation. For instance, the RG predefined interval lies between [-12.0, 12.0] dB. These intervals have been kept the same as the default values in the library.

The use of Generative Adversarial Networks (GANs) [21] for Deep Learning techniques has been tested by our team with less than satisfactory results. Beyond the overly complex process of training a GAN, the lack of constraints when sampling the learned space from the data can result in noisy and emotionally ambiguous samples.

We considered using SMOTE [22] to generate samples with some constraints but found that applying it to raw audio signal produced noisy samples due to the high dimensionality of the audio signal. Thus, we used the autoencoder applied for training the above-mentioned GAN to reduce the dimensionality of the sample, leading to a significant decrease in the number of values from around 482k to 60k, similar to the DeepSMOTE approach [23]. To the best of our knowledge, this was the first time this technique was applied to music samples.

It can be challenging to determine the best SMOTE implementation to use due to the many alternatives available. Kovács' article on SMOTE variants [24], along with the accompanying repository [7], provides a comprehensive resource for making a decision. The article compares over 80 variants, but we focused on the most widely used ones, namely SMOTE, BorderlineSMOTE, and Adasyn. After conducting preliminary tests, BorderlineSMOTE, particularly the Borderline_SMOTE2 implementation, was found to be the most suitable option.

3.7 Transfer Learning

One way to address the challenge of dealing with a small dataset is to leverage the knowledge learned from a domain with a larger data corpus. This involves transferring the learned weights from a pre-trained network to a new network with a different task. Our team has previously experimented with this approach by utilizing the learned weights of a network trained for genre recognition to improve emotion recognition. Instead of using a larger dataset, we took

[6] https://github.com/iver56/audiomentations

[7] https://github.com/analyticalmindsltd/smote_variants

Table 2. Datasets used for evaluation with respective sample distribution.

Dataset	Q1	Q2	Q3	Q4	Total
Original-4QAED	225	225	225	225	900
New-4QAED C	434	440	397	358	1629
New-4QAED B	343	343	343	343	1372

advantage of the learned information about genres, which are closely linked to specific emotion quadrants. For instance, heavy metal is typically associated with Q2, while reggae is connected to Q4 [25].

In the same direction, we conducted experiments to transfer the knowledge from the models introduced by Park et al. [8] in their article on artist classification. We opted for the simpler model in our work, which encompasses a series of 5 one-dimensional convolutional blocks, a global average pooling layer, and a dense layer that produces a 256-value vector. To carry out the experiment, we retrieved the model's weights from the article's accompanying repository [8] , loaded and froze them, and substituted the last layer with a dense layer that outputs to one of the quadrants.

We also evaluated the impact of utilizing data from larger music datasets using the available weights for the CRNN model on Won's repository trained on three datasets, but for this article we focus on MagnaTagATune (MTAT) [26] as it provided the best results. It should be noted that the CRNN model was trained for multi-label classification. The optimization process for the model was adaptive, which means that it changed during certain epochs of training. It started with Adam optimizer and a learning rate of 1e-3 until epoch 80, then shifted to the SGD optimizer with a learning rate of 1e-4, which decreased to 1e-5 at epoch 100, and finally to 1e-6 at epoch 120. According to the authors, this approach leads to a more stable training process and better results at 200 epochs. The model was optimized with a batch size of 16, which minimized the necessary computational resources for the training process.

4. EVALUATION DETAILS

As previously stated in Section 2, our team utilized the 4QAED dataset [9] for experimentation. The dataset, which consists of 900 samples, was evenly distributed across Russell's four quadrants. Each quadrant corresponds to a specific set of emotions: Q1 for happiness and excitement; Q2 for anger and frustration; Q3 for sadness and melancholy; and Q4 for serenity and contentment. The dataset provides 30-second excerpts of complete songs and two sets of emotionally relevant handcrafted features as data sources. The first set contains 1714 features found to be relevant for emotion recognition, while the second set contains the top 100 features obtained after feature selection. The dataset also provides categorical labels for one of the four quadrants as targets.

An expanded version of the dataset is also considered, increasing the number of available samples from 900 to 1629. The datasets are referred as Original-4QAED and New-4QAED, respectively. Table 2 shows the quadrant distribution of the datasets, including two variations of the New-4QAED dataset, a complete (C) version and a balanced subset (B) that has 1372 samples. The balanced subset takes into account the distribution of genre in each quadrant to prevent any potential bias.

4.1 Data Preprocessing

The input data for these methodologies were obtained through Mel-spectrogram representations generated using the Python library librosa [10] with default parameters. However, the sample rate was set to 16 kHz after experimenting with different values. This was done to reduce the computational complexity of the model and reduce the necessary resources for training and inference.

It is worth noting that DL-based architectures are robust to a lack of information related to lower sample rates when compared with higher rates, as has been observed in other studies [27].

4.2 Experimental Setup

The experiments were conducted on a server that was shared among the team. The server had two Intel Xeon Silver 4214 CPUs, which had a total of 48 cores and ran at a clock speed of 2.20GHz, and three NVIDIA Quadro P500 GPUs with 16GB of dedicated memory. The latter were necessary to develop and evaluate each network within a reasonable time frame. However, due to high demand, we also used Google Collaborator [11] during the evaluation process. Depending on availability, this platform offered a similar GPU and either an NVIDIA PCIE or an NVIDIA T4, both with 16GB of dedicated memory.

Almost all DL-based approaches were developed using the TensorFlow Python library [12] . This library enables the creation and optimization of intricate models in a straightforward and efficient manner. Additionally, in Section 3.7, we discussed how pre-trained CRNN models' provided weights were utilized with the PyTorch library [13] .

5. RESULTS AND DISCUSSION

In this section, we begin by describing the metrics considered and the evaluation strategy for the conducted experiments. The outcomes for each approach and datasets taken into consideration are presented in accordance with the broad categories discussed in Section 3.2. Refer to Table 3 for the comprehensive summary of the results.

In order to evaluate the performance of a classification model, three common metrics are Precision, Recall, and F1-score. Precision measures the proportion of true positive predictions within all positive predictions made by the model. Recall measures the proportion of true positive predictions within all actual positive samples. F1-score is a combined metric that takes into account both Precision

[8] https://github.com/jongpillee/ismir2018-artist
[9] Available at: http://mir.dei.uc.pt/resources/MER_audio_taffc_dataset.zip
[10] https://github.com/librosa/librosa
[11] https://colab.research.google.com/
[12] https://github.com/tensorflow/tensorflow
[13] https://pytorch.org/

and Recall, and is defined as the harmonic mean between the two. These metrics can be easily calculated using the scikit-learn library [14], which is widely used for machine learning tasks.

To obtain the metrics, we first optimized the relevant hyperparameters on Original-4QAED. We used a grid search strategy to experiment with a set of possible values to serve as a baseline for performance on New-4QAED. To ensure a fair comparison, we utilized the same parameters.

To ensure reliable results and handle small dataset sizes, a 10-fold and 10-repetition stratified cross-validation strategy is employed. This results in a total of 100 different train-test splits. In each repetition, the original dataset is randomly divided into ten portions while maintaining an equal distribution of quadrants. Nine of these portions are used for training, while the remaining one is used for testing. The portion held-out for testing changes for each train-test split, resulting in ten different combinations for each repetition.

The values of the hyperparameters that were tested using this approach varied depending on the methodology employed. In cases where the methodology was based on the baseline CNN, adjacent values were tested to accommodate any potential variations in the data. When the reported results did not have accompanying optimal hyperparameters values, they were used, and if not, the baseline CNN values were utilized as a fallback. However, regardless of the approach used, it is possible to draw meaningful conclusions concerning the effects of different dataset sizes and quadrant distributions.

Regarding improvements, increasing the dataset size had a positive impact on the performance of the baseline CNN with GRU and CRNN methodologies. The F1-score improved from 60.07% to 61.99% and 60.35% to 63.33%, respectively, when comparing the Original- to New-4QAED C datasets. This improvement was better compared to the DL Baseline. Additionally, the optimization phase was more stable with the increased dataset size. However, there was a slight decrease in performance when the balanced variations of the latter were applied, highlighting the importance of dataset size for optimal results.

Still, in terms of architecture improvements, it was found that the Hybrid Augmented methodology produced the best results. This particular methodology achieved an F1-score of 80.20% on the balanced subset of New-4QAED. The outcome was heavily influenced by both the size and quadrant distribution, with the latter potentially being linked to the biased nature of the DNN, much like traditional ML techniques.

It was also observed that using Time-Frequency Masking, Seven-Band Parametric Equalization, and Random Gain improved the results. These techniques achieved the best results with an increase of around 1.5% F1-score compared to the DL Baseline on Original-4QAED. They consistently performed better on New-4QAED as well. These findings suggest that more research is needed on data augmentation in MER, as most of the existing techniques are borrowed from other fields. As discussed in Section 1, the emotional impact of such techniques on the resulting samples is not yet fully understood.

All of the segment-level methodologies were found to perform similarly or worse than the DL Baseline. The poor performance may be attributed to the smaller size of the datasets used for training, compared to the ones used in the original proposal of the architectures. This means that the amount of available training data was limited, which may have hindered the performance of the models. Additionally, splitting the samples into smaller segments could have introduced more variability in the data, making it harder for the models to learn relevant features for distinguishing each quadrant. Further investigation is needed to verify this hypothesis.

The remaining methodologies related to knowledge transfer and data representation did not perform well compared to the baseline. The former significantly underperformed compared with the original implementations. This may imply that such approaches are not useful for emotion recognition, especially in the case of methodologies initially developed for multi-label classification, where larger datasets are used. The inadequate performance of these methodologies may be due to their significant differences from the learned features required for the task. As a result, crucial information for emotion recognition may be lost due to the abundance of irrelevant features. In the future, it may be worth experimenting with an ensemble of models trained for emotion recognition and other related tasks.

Finally, regarding embedding-based methodologies, we were not able to replicate the results presented for the OpenL3 embeddings on Original-4QAED, which was a 72% F1-score, reaching at most 55.70%. It seems that the unclear data splitting, where the authors followed an 80/10/10 train-validation-test data splitting instead of 10-fold cross-validation, might have contributed to this. Additionally, the original approach did not disclose the parameters used for creating the RF classifier, so we assumed default parameters from the scikit-learn implementation. To maintain consistency, we applied the usual method for cross-validation. As for the autoencoder embeddings, these exhibited better performance on New-4QAED overall than OpenL3 embeddings. This observation suggests that OpenL3 embeddings may not be the most appropriate choice for MER.

Furthermore, the DeepSMOTE-based augmentation did not show any significant improvement over the DL baseline. The reason behind this lack of improvement could be the high dimensional embedding space, which provides little variability when sampled as compared to the original samples. To overcome this, it might be helpful to reduce the input data size by using the segments of the whole samples. This should decrease the embedding space dimension and provide more relevant synthesized samples.

6. CONCLUSION AND FUTURE DIRECTIONS

The study aimed to compare the effectiveness of ML and DL methodologies for static emotion recognition in music with audio with datasets of different sizes. The fo-

[14] https://scikit-learn.org/stable/

Table 3. Datasets used for evaluation with respective sample distribution.

Methods	Original-4QAED			New-4QAED C			New-4QAED B		
	P	R	F1	P	R	F1	P	R	F1
SVM Baseline	75.63%	76.03%	75.59%	69.92%	70.26%	69.79%	70.03%	70.05%	69.82%
DL Baseline	61.60%	61.21%	60.62%	62.46%	63.99%	61.66%	61.39%	63.42%	60.28%
Baseline with GRU	61.58%	61.01%	60.07%	62.29%	62.46%	61.99%	60.69%	60.01%	58.85%
CRNN	**65.14%**	**65.07%**	**64.63%**	**64.20%**	**64.03%**	**64.09%**	**63.31%**	**63.34%**	**62.54%**
Hybrid Augmented	**67.81%**	**68.08%**	**68.04%**	**68.15%**	**68.14%**	**67.85%**	***80.56%***	***80.50%***	***80.24%***
ShortChunk CNN	**64.66%**	61.48%	60.61%	**64.07%**	62.13%	61.84%	60.23%	59.19%	57.07%
Sample CNN	**62.64%**	61.26%	60.92%	**65.17%**	62.62%	60.78%	**62.43%**	56.70%	54.46%
OpenL3	55.67%	56.75%	55.70%	53.92%	54.49%	53.62%	53.03%	53.18%	52.85%
Autoencoder	50.63%	50.40%	50.18%	53.78%	55.45%	53.56%	53.56%	54.76%	53.69%
Baseline + TFM	**63.05%**	**62.75%**	**62.03%**	62.51%	62.17%	61.82%	62.33%	61.85%	61.39%
Baseline + SB	**63.38%**	**62.79%**	**62.12%**	62.54%	62.16%	61.73%	62.13%	61.71%	61.01%
Baseline + RG	**63.37%**	**63.13%**	**62.24%**	63.02%	62.80%	62.08%	62.35%	62.10%	61.36%
Baseline + DeepSMOTE	61.91%	61.61%	60.70%	62.40%	62.02%	61.47%	61.62%	61.62%	60.48%
Artists CNN	51.95%	53.93%	50.85%	51.81%	53.29%	50.27%	51.56%	52.43%	50.22%
CRNN Pre-trained MTAT	51.93%	51.71%	50.21%	52.97%	53.72%	51.70%	52.16%	52.50%	51.44%

cus was more on DL to address the semantic gap found in traditional ML approaches by exploring various approaches, such as improving existing architectures, using segment-level models, applying data augmentation, performing knowledge transfer, and using different data representations as input.

The proposed Hybrid Augmented achieved the best result overall on the New-4QAED balanced dataset with an 80.20% F1 Score. This methodology is an ensemble of a CNN trained with additional synthesized samples and a DNN, which use Mel-spectrogram representations and previously extracted features from each song as input, respectively. Another notable improvement was observed by using CRNN on the increased size of the New-4QAED datasets. This approach outperformed the DL Baseline by around 2% on the complete set, outperforming methodologies that applied classical data augmentation techniques.

When comparing different methodologies, it was found that classical audio augmentation techniques and architectural improvements effectively improved performance. However, segment-level architectures, knowledge transfer from related tasks, and embedding-based input representations did not show as much improvement. Nevertheless, as discussed earlier, there is room for improvement in these areas. The results also revealed that, in most cases, the size of the dataset has a greater impact on classification performance than class balance.

The findings suggest that there is a need for continued research aimed at developing new classical features and enhancing DL architectures for improved performance. Additionally, exploring data augmentation techniques specifically for MER could be a promising approach to fully leverage DL models' ability to automatically extract relevant features. As more training data becomes available, future DL architectures should include an RNN component to capture time-domain-specific features. Finally, using various spectral representations as inputs is an exciting area for further research, as demonstrated by early experimental studies. However, it is important to address the unstable nature of these approaches before they can be fully utilized.

Acknowledgments

This work is funded by FCT - Foundation for Science and Technology, I.P., within the scope of the projects: MERGE - PTDC/CCI-COM/3171/2021 financed with national funds (PIDDAC) via the Portuguese State Budget; and project CISUC - UID/CEC/00326/2020 with funds from the European Social Fund, through the Regional Operational Program Centro 2020. Renato Panda was supported by Ci2 - FCT UIDP/05567/2020.

7. REFERENCES

[1] Y. Feng, Y. Zhuang, and Y. Pan, "Popular music retrieval by detecting mood," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 375–376.

[2] L. Lu, D. Liu, and H.-J. Zhang, "Automatic mood detection and tracking of music audio signals." *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 5–18, 2006.

[3] R. Panda, R. Malheiro, and R. P. Paiva, "Novel audio features for music emotion recognition," *IEEE Transactions on Affective Computing*, vol. 11, pp. 614–626, 2020.

[4] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Convolutional recurrent neural networks for music classification," in *Proceedings of the 2017 International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 2392–2396.

[5] M. Malik, S. Adavanne, K. Drossos, T. Virtanen, D. Ticha, and R. Jarina, "Stacked convolutional and

recurrent neural networks for music emotion recognition," in *Proceedings of the 14th Sound and Music Computing Conference*, vol. 14, 2017, pp. 208–213.

[6] J. Lee, J. Park, K. L. Kim, and J. Nam, "Sample-level deep convolutional neural networks for music auto-tagging using raw waveforms," in *Proceedings of the 14th Sound and Music Computing Conference*, vol. 14, 2017, pp. 220–226.

[7] J. Pons, O. Nieto, M. Prockup, E. M. Schmidt, A. F. Ehmann, and X. Serra, "End-to-end learning for music audio tagging at scale," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 637–644.

[8] J. Park, J. Lee, J. Park, J.-W. Ha, and J. Nam, "Representation learning of music using artist labels," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 717–724.

[9] E. Koh and S. Dubnov, "Comparison and analysis of deep audio embeddings for music emotion recognition," 2021. [Online]. Available: https://arxiv.org/abs/2104.06517

[10] R. Mignot and G. Peeters, "An analysis of the effect of data augmentation methods: Experiments for a musical genre classification task," *Transactions of the International Society for Music Information Retrieval*, vol. 2, pp. 97–110, 2019.

[11] K. Choi, G. Fazekas, and M. Sandler, "Explaining deep convolutional neural networks on music classification," 2016. [Online]. Available: https://arxiv.org/abs/1607.02444

[12] M. Won, S. Chun, and X. Serra, "Toward interpretable music tagging with self-attention," 2019. [Online]. Available: https://arxiv.org/abs/1906.04972

[13] A. Pannese, M.-A. Rappaz, and D. Grandjean, "Metaphor and music emotion: Ancient views and future directions," *Consciousness and Cognition*, vol. 44, pp. 61–71, 2016.

[14] K. Hevner, "Experimental studies of the elements of expression in music," *The American Journal of Psychology*, vol. 48, pp. 246–268, 1936.

[15] J. Posner, J. A. Russell, and B. S. Peterson, "The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology," *Development and Psychopathology*, vol. 17, pp. 715–734, 2005.

[16] T. Eerola and J. K. Vuoskoski, "A comparison of the discrete and dimensional models of emotion in music," *Psychology of Music*, vol. 39, pp. 18–49, 2011.

[17] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," in *Proceedings of the 17th International Society for Music Information Retrieval Conference*, 2016, pp. 805–811.

[18] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," in *Proceedings of the 17th Sound and Music Computing Conference*, 2020, pp. 331–337.

[19] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014. [Online]. Available: https://arxiv.org/abs/1409.1259

[20] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Proceedings of the 20th Annual Conference of the International Speech Communication Association*, 2019, pp. 2613–2617.

[21] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[22] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: Synthetic minority oversampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[23] D. Dablain, B. Krawczyk, and N. V. Chawla, "Deepsmote: Fusing deep learning and smote for imbalanced data," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 16, pp. 1–15, 2022.

[24] G. Kovacs, "An empirical comparison and evaluation of minority oversampling techniques on a large number of imbalanced datasets," *Applied Soft Computing*, vol. 83, p. 105662, 2019.

[25] D. Griffiths, S. Cunningham, J. Weinel, and R. Picking, "A multi-genre model for music emotion recognition using linear regressors," *Journal of New Music Research*, vol. 50, pp. 355–372, 2021.

[26] E. Law, K. West, M. I. Mandel, M. Bay, and J. S. Downie, "Evaluation of algorithms using games: The case of music tagging," in *Proceedings of the 10th International Society for Music Information Retrieval Conference*, vol. 50, 2009, pp. 387–392.

[27] K. Pyrovolakis, P. Tzouveli, and G. Stamou, "Multimodal song mood detection with deep learning," *Sensors*, vol. 22, pp. 387–392, 2009.

CHORDSYNC: CONFORMER-BASED ALIGNMENT OF CHORD ANNOTATIONS TO MUSIC AUDIO

Andrea POLTRONIERI (andrea.poltronieri2@unibo.it) (0000-0003-3848-7574) [1],
Valentina PRESUTTI (valentina.presutti@unibo.it) (0000-0002-9380-5160) [1], and
Martín ROCAMORA (martin.rocamora@upf.edu) (0000-0003-3183-9717) [2,3]

[1] **University of Bologna**, Bologna, Italy
[2] **Universitat Pompeu Fabra**, Barcelona, Spain
[3] **Universidad de la República**, Montevideo, Uruguay

ABSTRACT

In the Western music tradition, chords are the main constituent components of harmony, a fundamental dimension of music. Despite its relevance for several Music Information Retrieval (MIR) tasks, chord-annotated audio datasets are limited and need more diversity. One way to improve those resources is to leverage the large number of chord annotations available online, but this requires aligning them with music audio. However, existing audio-to-score alignment techniques, which typically rely on Dynamic Time Warping (DTW), fail to address this challenge, as they require weakly aligned data for precise synchronisation. In this paper, we introduce *ChordSync*, a novel conformer-based model designed to seamlessly align chord annotations with audio, eliminating the need for weak alignment. We also provide a pre-trained model and a user-friendly library, enabling users to synchronise chord annotations with audio tracks effortlessly. In this way, ChordSync creates opportunities for harnessing crowd-sourced chord data for MIR, especially in audio chord estimation, thereby facilitating the generation of novel datasets. Additionally, our system extends its utility to music education, enhancing music learning experiences by providing accurately aligned annotations, thus enabling learners to engage in synchronised musical practices.

1. INTRODUCTION

Harmony is central to Western music traditions' theoretical and practical foundations. It entails the combination of individual pitches to create chords and their concatenation into sequences to create chord progressions. Therefore, chords, i.e., the simultaneous sounding of two or more pitches, are the primary constituents of harmony, while chord progressions play a vital role in shaping and defining the overall structure of a musical piece.

Not surprisingly, automatic chord recognition (ACR) from audio, the task of generating a sequence of time-synchronised chord labels given raw audio as input, has been an active research topic in Music Information Retrieval (MIR) for more than two decades [1], with applications including music similarity assessment [2, 3], classification [4], and segmentation [5].

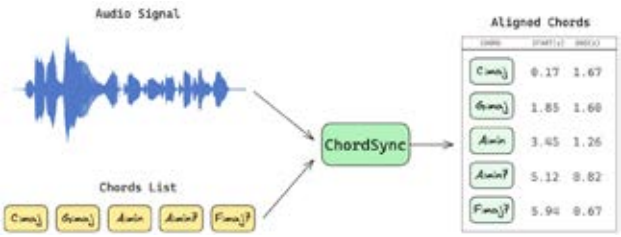


Figure 1. Basic schema of ChordSync: The model processes a list of chords alongside the audio signal, producing time-aligned chords as output.

The development of ACR systems requires large datasets of audio-aligned chord annotations for training and evaluation. However, the diversity of existing chord annotated datasets is limited. They predominantly feature pop music and exclude a wide array of genres and styles [1]. The lack of diversity is critical since the chord vocabulary differs according to musical style and context, making it difficult to generalise from a limited music sample. Besides, the subjectivity inherent in chord annotations further complicates the ACR task. Musical chords can be annotated at varying levels of granularity and complexity, accounting for global harmony or specific instrument contributions. Additionally, the distinction between harmony and melodic lines is frequently challenging, while interpretations of elements such as arpeggios often lead to divergent annotations. In [6], authors demonstrate that inter-annotator agreement on the root note in a dataset annotated by four different annotators stands at only 76%. Datasets annotated from other perspectives are even rarer, currently comprising only a few dozen tracks.

In recent years, meta-corpora of chord annotations have emerged, such as Chord Corpus (ChoCo) [7] and When in Rome (WiR) [8], which aim to aggregate and standardise different datasets originally available in various formats and annotation styles. In this way, they facilitate the utilisation of large-scale data, which improves diversity and is crucial for training deep-learning models. However, the availability of audio-aligned annotations within these cor-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

pora remains limited. Notably, less than 12% of the 20,000 annotated tracks in ChoCo are audio-aligned.

On the other hand, the internet hosts vast repositories of crowd-sourced chord annotations on platforms such as Ultimate Guitar [1], e-chords [2], and Chordie [3], collectively housing millions of annotated songs. This multitude offers a great variety in terms of genre distribution, including genres not present in any MIR datasets, such as electronic, metal, hip hop, reggae, and country. Moreover, these repositories of harmonic annotations often contain multiple versions of the same song. This abundance of versions may offer new avenues for analysis, accommodating the subjectivity and complexity inherent in the annotations, as proposed in [6,9]. Unfortunately, these annotations lack any timing and duration information, providing solely lists of chords and occasionally lyrics, hindering their reuse for MIR-related tasks.

These challenges underscore the need for systems capable of aligning chord annotations with audio recordings. Yet, to the best of our knowledge, no model has been explicitly developed for this purpose. Existing audio-to-score alignment techniques often rely on Dynamic Time Warping (DTW) algorithms [10], typically requiring preliminary weak alignment. Such alignment methods are not always feasible for aligning chord annotations to audio, particularly in cases of crowd-sourced data where temporal information is completely lacking.

1.1 Our Contribution

In this paper, we address this gap by introducing *ChordSync*, a novel approach that seamlessly aligns chord annotations to audio without requiring any preliminary weak alignment (see Figure 1). Leveraging the power of conformer architecture [11], our method paves the way for creating diverse and comprehensive audio-aligned chord annotated datasets based on existing resources. We also provide a pre-trained model and a user-friendly library, enabling users to synchronise chord annotations with audio tracks effortlessly. Finally, we showcase the effectiveness of our approach by aligning a sample of tracks taken from Ultimate Guitar. This can, in turn, benefit other MIR applications, such as music structure analysis, and foster enriched music learning experiences.

The rest of the paper is structured into four main sections: Section 2 reviews the current state-of-the-art, Section 3 describes the methodology of *ChordSync*, Section 4 presents experimental results, and Section 5 offers conclusions and suggests future research directions.

2. RELATED WORK

2.1 Audio-to-Score Alignment

The task of aligning audio to symbolic music, commonly known as *audio-to-score alignment* (A2SA), has been primarily addressed by *Dynamic Time Warping (DTW)* algorithms [12], as they are particularly effective for sequence alignment tasks. Thus, various DTW-based alignment methods have been proposed to align audio with different symbolic music formats, such as MIDI [13], often integrating additional techniques and diverse signal representations to improve alignment accuracy [14, 15].

A differentiable variant of DTW, *SoftDTW*, has been recently used as the loss function within neural network architectures, mainly for multi-pitch estimation tasks [16, 17]. However, a general limitation of the DTW-based approaches is their reliance on weak-aligned data to perform the alignment. This requirement renders them unsuitable for contexts without prior alignment information.

Other deep-learning methods have been investigated for audio-to-score alignment, including leveraging automatic transcription techniques [18] and training audio features tailored explicitly for alignment tasks [19].

The only previously proposed approach for aligning audio with chord annotations uses Hidden Markov Models (HMM) and is part of an ACR workflow [20]. Also related to our work is the *Harmonic Change Detector (HCD)*, introduced in [21] and subsequently revisited and improved in [22, 23], for detecting harmonic changes within the audio signal, including chord changes. However, the number of harmonic changes within the audio signal often exceeds the number of chord changes, posing challenges for using these algorithms directly for audio-to-chord alignment.

2.2 Lyrics-to-Audio Alignment

Another form of alignment pertinent to our work is the audio-to-lyrics alignment task, which seeks to determine the corresponding locations in a song recording of its lyrics at various levels such as line, word, or phoneme [24]. Existing methods for this task are commonly adapted from automatic speech recognition (ASR) [25, 26], despite the inherent complexity of singing voices compared to speech [27], and typically make use of acoustic models trained to recognise the phonetic content of the audio signal at various levels of granularity. Some recent works have adopted the Connectionist Temporal Classification (CTC) loss [28], training the acoustic model in an end-to-end fashion [26].

2.3 Conformer-based Approaches

The conformer architecture [11] has recently emerged in ASR as a novel architecture to effectively model global and local audio dependencies by leveraging a combination of Convolutional Neural Networks (CNNs) and Transformer architectures. It has showcased remarkable success across various tasks not only in speech [29] but also in music [30], including melodic transcription [31], representation learning [32], and music audio enhancement [33].

3. METHOD

This section describes *ChordSync*, our proposed conformer-based model for audio-to-chord alignment. It implements an acoustic model for estimating the frame-wise probabilities of chord labels, which are then fed to a forced-alignment decoder, along with the list of chord labels to align. Figure 2 illustrates the three primary

[1] Ultimate Guitar: `https://www.ultimate-guitar.com/`
[2] e-chords: `https://www.e-chords.com/`
[3] Chordie: `https://www.chordie.com/`

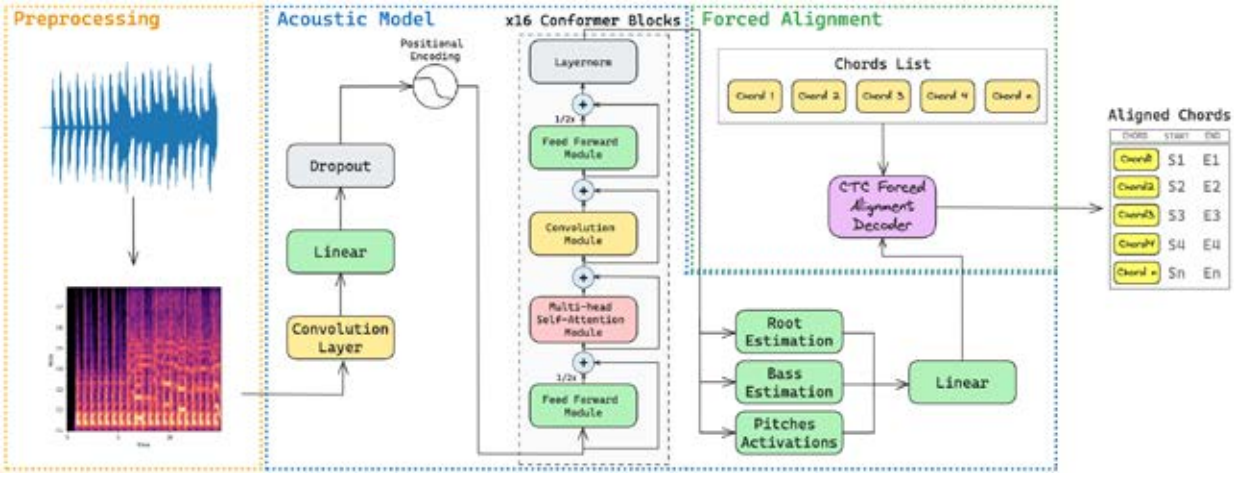


Figure 2. Architecture of *ChordSync*: (i) The audio signal undergoes preprocessing to Constant-Q Transform (yellow box); (ii) The preprocessed audio serves as input for training the conformer-based acoustic model (blue box); and (iii) The model output probabilities, along with the list of chord labels for alignment, is fed into a CTC forced alignment module (green box), which outputs the aligned chord labels.

steps implemented by the model: pre-processing and data augmentation (Section 3.2), the acoustic model used during training (Section 3.3), and the forced alignment decoder (Section 3.4). The software implementation and a pre-trained model are available on a GitHub repository. [4]

3.1 Problem Statement

Let $X = \{x_1, ..., x_N\}$ be a frame-level sequence of acoustic features extracted from the input audio, where $x_n \in \mathbb{R}^D$ represents a D-dimensional feature vector, and N indicates the total number of frames within the sequence. Let $C = \{c_1, ..., c_M\}$ be the input list of chord labels encoded into integer values, where $c_m \in \mathbb{Z}^K$, K denotes the size of the chord vocabulary, and M is the length of the chord sequence. The list of chord labels is upsampled to match the length of the audio sequence N. This upsampling is performed uniformly, assuming each chord has a duration approximately equal to N/M. Specifically, each chord label c_m is repeated for approximately N/M frames to produce the sequence $Z = \{z_1, ..., z_N\}$, where $z_m \in \mathbb{Z}^K$. Thus, we train an acoustic model to optimise the following equation:

$$Z^* = \underset{z}{\operatorname{argmax}}\, p(Z|X), \tag{1}$$

where Z^* represents the optimal sequence of chord labels that maximises the posterior probability $p(Z|X)$, given the input sequence X. Note that X and Z are aligned at the frame level, and $p(X|Z)$ is evaluated by estimating the frame-wise posterior probability $p(x_n|z_n)$.

The output probabilities $p(X|Z)$ from the acoustic model are then fed to a CTC forced alignment decoder, which estimates the best alignment between the sequence of acoustic features X and the list of chord labels C:

$$A^* = \underset{a}{\operatorname{argmax}}\, p(A|X, C), \tag{2}$$

where A^* represents the optimal alignment between X and C that maximises the posterior probability $p(A|X, C)$.

In this way, the decoder generates the aligned chord labels with respect to the audio signal.

3.2 Preprocessing

For the input audio data, a standard pre-processing pipeline is implemented. The audio is first resampled to a sampling rate of 22050 Hz, and a hop size of 2048 is applied. Then, the Constant-Q Transform (CQT) features are calculated on 6 octaves starting from $C1$, with 24 bins per octave, resulting in a total of 144 bins.

The audio data used for training undergoes data augmentation by applying (i) time masking and (ii) frequency masking directly to the audio features, as proposed in *SpecAugment* for end-to-end ASR [34].

During training, each audio excerpt in the training set undergoes augmentation, where either one of the transformations (frequency masking or time masking) or both are applied, and the choice of augmentation technique is determined randomly with equal probability.

Chord labels are numerically encoded into integer values and upsampled to match the length of the audio sequence N. The upsampling is performed using the `pumpp` library [5]. Figure 3 shows how chord labels are converted and sampled. The size of the chord vocabulary K results from the linear combination of the 12 pitches, representing the chromatic scale, with chord qualities such as {`maj`, `min`, `7`, `dim`, `dim7`, `hdim7`, `aug`, `min7`, `maj7`, `maj6`, `min6`, `minmaj7`, `sus2`, `sus4`}, plus an additional chord symbol N representing silence or no chord.

3.3 Conformer-based Acoustic Model

The acoustic model we adopt is an adaptation of the original Conformer architecture [11], where the audio encoder

[4] `https://github.com/andreamust/ChordSync`

[5] `https://github.com/bmcfee/pumpp`.

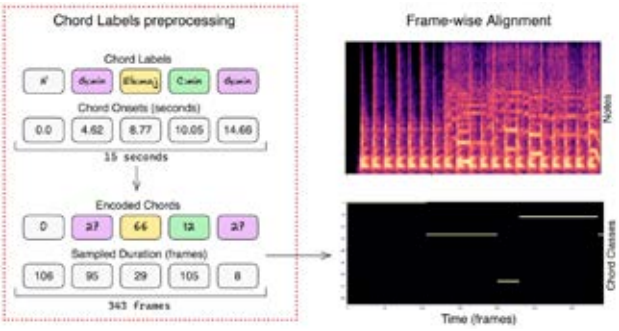


Figure 3. Workflow of the pre-processing applied to the chord labels. Chord labels are numerically encoded and upsampled to match the length of the CQT.

processes the input through a convolutional module followed by a series of Conformer blocks.

The convolutional module comprises a convolution layer, a fully connected layer, and a dropout layer. The convolutional module serves as the initial feature extractor, capturing local patterns within the input CQT. Dropout regularisation is applied by randomly deactivating units during training to reduce overfitting. Additionally, we incorporate positional encoding, as proposed in the original transformer architecture paper [35].

A Conformer block is composed of four modules stacked together: a feed-forward module, a self-attention module, a convolution module, and a second feed-forward module at the end. In the original Conformer paper, the authors explore three different sizes of the Conformer architecture: S (small), M (medium), and L (large), with different numbers of layers, hidden units, and other parameters. For our implementation, we opt for the M architecture, which comprises a 16 encoder layer with a dimension of 256, 4 attention heads, and a convolutional kernel size of 32. While the original paper observed significant improvements when transitioning from the S to the M variant, our experimentation yielded little improvements from M to L.

To handle the large dimensionality of the vocabulary, we use an architecture similar to that proposed by [36], in which root notes, bass notes, and all pitch activations of the chord are predicted. Subsequently, these probabilities are passed to a feed-forward layer, which converts these three probabilities into the likelihood of the chord with respect to the vocabulary K, similarly to what was proposed by [37].

For training, we employ cross-entropy loss and optimise using the AdamW optimiser. We utilise a cosine annealed warm restart learning scheduler to manage learning rates. Learning rate schedulers proved effective in training audio data, especially with augmented data [34]. Finally, we applied early stopping by halting the training if the loss failed to decrease for over 20 epochs to prevent overfitting.

3.4 Forced Alignment

To estimate the best alignment between the acoustic features X and the chord labels C, we utilise the Connectionist Temporal Classification (CTC) objective function [28], which computes the probability of a given alignment between the input features and output labels. The CTC objective function is defined as follows:

$$p(C|X) = \sum_{A \in \mathcal{A}_{X,C}} p(a_t|X), \tag{3}$$

where $\mathcal{A}_{X,C}$ denotes the set of all possible alignments that produce the label sequence C, and $p(a_t|X)$ represents the probability of alignment a_t given the input sequence X.

The probability of alignment a_t given X is computed as the sum of probabilities of all paths a'_t that correspond to a_t after collapsing repeated labels and blank symbols:

$$p(a_t|X) = \prod_{t=1}^{T} p_t(\pi_t|X), \tag{4}$$

where T is the length of the alignment, and $p(\pi_t|X)$ is the probability of the t-th symbol in the alignment path π given the input sequence X.

4. EXPERIMENTS

Due to the lack of established methodologies to address the chord-to-score alignment task, conducting a comparative evaluation with existing state-of-the-art techniques presents some challenges. Therefore, to gauge the effectiveness of the proposed methodologies, we use alternative approaches performing analogous albeit slightly dissimilar methods for comparison and conduct two different experiments. The first aims to evaluate the model's capability in detecting chord boundaries, while the second compares it to a traditional DTW-based alignment.

All experiments were carried out using a subset of ChoCo [7], which offers a standardised version of chord annotations sourced from various datasets. Specifically, only ChoCo partitions annotated on audio, i.e. expressing temporal information such as onsets and duration in seconds, were considered. Table 1 presents a summary of all ChoCo partitions employed for training and evaluation.

Audio files corresponding to each ChoCo annotation were obtained automatically from the available metadata in the original datasets. This was necessary as only a small portion of the datasets offer external links to the original audio sources used for chord annotation. Since the automatic retrieval process depends on sometimes sparse and

Dataset	Genre	#Tracks	Reference
Isophonics	pop, rock	300	[38]
Billboard	pop	740	[39]
Chordify	pop	50	[40]
Robbie Williams	pop	61	[41]
Uspop 2002	pop	195	[42]
RWC-Pop	pop	100	[43]
Schubert-Winterreise	classical	225	[44]
Weimar Jazz Database	jazz	456	[45]
JAAH	jazz	113	[46]
Total		**2240**	

Table 1. Dataset utilised for all experiments. All datasets are sourced from ChoCo [7].

Method	Genre	Precision ↑	Recall ↑	F1 Score ↑
HCDF	pop/rock	0.4999	0.6334	0.5269
HCDF	classical	0.4454	0.6220	0.5191
HCDF	jazz	0.4911	0.7749	0.5857
HCDF	all	0.4953	0.6508	0.5323
ChordSync	pop/rock	0.8847	0.8335	0.8553
ChordSync	classical	0.6008	0.5917	0.5951
ChordSync	jazz	0.4663	0.4129	0.4350
ChordSync	all	**0.8895**	**0.8420**	**0.8621**

Table 2. Precision, Recall, and F1 Score for the *HCDF* method [23] and the proposed *ChordSync* model.

incomplete metadata, the validity of the audio files was manually verified on randomly selected samples. The complete dataset consists of 2240 audio tracks, encompassing four distinct music genres: pop, rock, classical, and jazz. However, it is noteworthy to observe a significant imbalance in the dataset, with the pop/rock genre comprising over 65% of the total tracks.

Audio data is segmented into intervals of 15 seconds duration, with a 3-second overlap between each segment and the preceding one, yielding a corpus of 31909 segments. We split these segments into train, validation, and test sets with proportions of $65-20-15$. Importantly, when a segment from a particular song is included in the train set, we ensure that no segments from the same song are included in either the validation or test sets.

4.1 Chord Changes Detection Evaluation

The first comparison is conducted with the Harmonic Change Detection algorithm [21], which specialises in detecting harmonic changes on an audio signal. These algorithms are typically evaluated by assessing their capacity to detect the onsets of annotated chords within the identified harmonic changes, often employing standard metrics such as Precision, Recall, and F1 Score.

However, by their intrinsic design, HCD algorithms extract every harmonic variation present in the audio signal. [21] and [23] provide two distinct implementations of this algorithm, each optimising either the F1 score or precision. The number of harmonic changes varies significantly depending on the chosen algorithm implementation, but in general, it far exceeds that of chord changes.

In contrast, *ChordSync* extracts the number of chord changes of the list of chords passed to the CTC decoder. Table 2 presents a comparative analysis between the HCD algorithm in [23] and *ChordSync*. A harmonic change match is defined in a 0.3 seconds window between the predicted and the ground-truth onsets.

Our method demonstrates notable efficacy in chord change extraction, substantially increasing all the performance measures considered. This performance improvement stems from the model's inherent design, which optimises the alignment between the audio signal and the provided sequence of chords. However, performance decreases in the less represented genres within the dataset, such as jazz and classical.

4.2 Alignment Evaluation

Evaluating audio-to-score or audio-to-lyrics alignment entails comparing predicted and ground truth timestamps to measure their temporal differences [47, 48]. This comparison typically occurs pairwise and involves calculating metrics such as the median absolute error in seconds and the percentage of overlapping segments. This approach offers a straightforward means of assessing alignment accuracy and determining the effectiveness of alignment methods for practical applications.

Furthermore, perceptually-grounded metrics for evaluating lyrics-to-audio alignment systems have been recently introduced [49]. These metrics were fine-tuned on data collected through a user Karaoke-like experiment, reflecting human judgement of how "synchronous" lyrics and audio stimuli are perceived in that setup.

All the metrics described above are implemented in the `mir_eval` library [50], providing a standardised and accessible means for conducting evaluations in audio alignment. Given its similarities with other alignment tasks and the perceptual considerations involved, the same metrics are suitable for evaluating audio-to-chord alignment.

We compare the performance of *ChordSync* and a conventional DTW-based approach using the *SyncToolbox* library [10], which offers a diverse array of DTW-based implementations. The evaluation of this type of approach requires both symbolic sequences weakly aligned to audio, which are a prerequisite for the alignment, and ground truth annotations strong aligned to audio for evaluation. To our current knowledge, such annotations are exclusively found within the Schubert Winterreise dataset [44]. Consequently, the evaluation of this approach is constrained to a limited number of pieces and to the *classical* genre.

To perform the alignment between audio and chord annotations, the chord annotations were first decomposed into their constituent notes, each of which was then associated with the chord's symbolic onsets. The audio data underwent pre-processing using chroma and DLNCO features, known for their effectiveness in alignment tasks [51]. Finally, alignment was carried out utilising memory-restricted multi-scale DTW (MrMsDTW) [52, 53].

Table 3 shows the performance of the proposed model on the Schubert Winterreise dataset compared to a standard DTW approach, along with the broader performance metrics of the ChordSync method applied across all datasets (c.f. Table 1). This evaluation demonstrates that the proposed model accurately detects chord changes and achieves alignment performance comparable to that of a DTW-based approach. Conversely, the evaluation conducted solely on a subset of the Winterreise dataset demonstrates performance comparable to DTW, albeit slightly lower. However, this data highlights the model's strong generalisation capabilities, as it effectively aligns songs from a genre that was statistically rare in the training data due to its limited size.

Even so, it is worth noting that the proposed model achieves these results without relying on weak-aligned data, which is a requirement for DTW-based approaches.

Method	Dataset	Percentage Correct ↑	Median Absolute Error ↓	Average Absolute Error ↓	Perceptual ↑
DTW	schubert-winterreise	0.8621	0.0661	0.2088	0.7895
ChordSync	schubert-winterreise	0.8245	0.2641	0.2512	0.7230
ChordSync	all	0.8664	0.4224	0.5001	0.7900

Table 3. Performance of ChordSync on the Schubert-Winterreise dataset [44] compared to a standard DTW approach performed using the *SyncToolbox* library (first two rows). Additionally, performance metrics of the ChordSync method applied across all datasets are presented. Metrics are computed with the alignment metrics from the `mir_eval` library.

5. DISCUSSION AND CONCLUSION

In this paper, we introduce *ChordSync*, a novel model for audio-to-chord alignment that harnesses the capabilities of the Conformer architecture [11]. Our proposed method attains performance levels comparable to DTW algorithms in the audio-to-chord alignment task without requiring any pre-existing alignment as in the DTW approaches. Therefore, our method facilitates the creation of diverse and comprehensive datasets featuring synchronised audio and chord annotations by exploiting existing resources, such as crowd-sourced online chord annotations, which typically lack timing and duration information. In order to do that, we offer a pre-trained model and a user-friendly library, empowering users to synchronise chord annotations with audio tracks effortlessly.

The primary limitation of the proposed approach stems from its reliance on an acoustic model trained using a simplified vocabulary of chord labels (see Section 3) because the model's performance is contingent upon the vocabulary size. If a chord is absent from the chord vocabulary, it will inevitably be approximated by the existing label of the most similar chord in the vocabulary. However, if the two consecutive chord symbols match, the alignment gets more challenging for the CTC decoder. Although the results indicate that the decoder can handle such scenarios, using alternative chord encoding might yield better performance.

Furthermore, investigating alternative chord encoding could make the model key-agnostic, a feature lacking in the current model, which is not specifically designed to handle discrepancies in key between the chord labels and the audio signal.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101004746.

6. REFERENCES

[1] J. Pauwels, K. O'Hanlon, E. Gómez, and M. B. Sandler, "20 years of automatic chord recognition from audio," in *Proceedings of the 20th International Society for Music Information Retrieval Conference, ISMIR 2019, Delft, The Netherlands, November 4-8, 2019*, A. Flexer, G. Peeters, J. Urbano, and A. Volk, Eds., 2019, pp. 54–63.

[2] W. B. de Haas, F. Wiering, and R. C. Veltkamp, "A geometrical distance measure for determining the similarity of musical harmony," *Int. J. Multim. Inf. Retr.*, vol. 2, no. 3, pp. 189–202, 2013.

[3] J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "The harmonic memory: a knowledge graph of harmonic patterns as a trustworthy framework for computational creativity," in *Proceedings of the ACM Web Conference 2023, WWW 2023, Austin, TX, USA, 30 April 2023 - 4 May 2023*, Y. Ding, J. Tang, J. F. Sequeda, L. Aroyo, C. Castillo, and G. Houben, Eds. ACM, 2023, pp. 3873–3882.

[4] Y. Huang, S. Lin, H. Wu, and Y. Li, "Music genre classification based on local feature selection using a self-adaptive harmony search algorithm," *Data Knowl. Eng.*, vol. 92, pp. 60–76, 2014.

[5] J. Pauwels, F. Kaiser, and G. Peeters, "Combining harmony-based and novelty-based approaches for structural segmentation," in *International Society for Music Information Retrieval Conference*, 2013.

[6] H. V. Koops, W. B. De Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, vol. 48, no. 3, p. 232–252, may 2019.

[7] J. de Berardinis, A. Meroño-Peñuela, A. Poltronieri, and V. Presutti, "Choco: a chord corpus and a data transformation workflow for musical harmony knowledge graphs," *Scientific Data*, vol. 10, no. 1, p. 641, Sep 2023.

[8] M. Gotham, G. Micchi, N. N. López, and M. Sailor, "When in rome: a meta-corpus of functional harmony," *Transactions of the International Society for Music Information Retrieval*, vol. 6, no. 1, 2023.

[9] H. V. Koops, W. B. de Haas, J. Bransen, and A. Volk, "Chord label personalization through deep learning of integrated harmonic interval-based representations," *CoRR*, vol. abs/1706.09552, 2017. [Online]. Available: http://arxiv.org/abs/1706.09552

[10] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync toolbox: A python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software*, vol. 6, no. 64, p. 3434, 2021.

[11] A. Gulati, J. Qin, C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for

speech recognition," in *Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event, Shanghai, China, 25-29 October 2020*, H. Meng, B. Xu, and T. F. Zheng, Eds. ISCA, 2020, pp. 5036–5040.

[12] A. Morsi and X. Serra, "Bottlenecks and solutions for audio to score alignment research," in *Proceedings of the 23rd International Society for Music Information Retrieval Conference, ISMIR 2022, Bengaluru, India, December 4-8, 2022*, P. Rao, H. A. Murthy, A. Srinivasamurthy, R. M. Bittner, R. C. Repetto, M. Goto, X. Serra, and M. Miron, Eds., 2022, pp. 272–279.

[13] C. Raffel and D. P. W. Ellis, "Optimizing dtw-based audio-to-midi alignment and matching," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 81–85.

[14] J. J. Carabias-Orti, F. J. Rodríguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada, "An audio to score alignment framework using spectral factorization and dynamic time warping," in *Proceedings of the 16th International Society for Music Information Retrieval Conference, ISMIR 2015, Málaga, Spain, October 26-30, 2015*, M. Müller and F. Wiering, Eds., 2015, pp. 742–748.

[15] F. J. Rodriguez-Serrano, J. J. Carabias-Orti, P. Vera-Candeas, and D. Martinez-Munoz, "Tempo driven audio-to-score alignment using spectral decomposition and online dynamic time warping," *ACM Trans. Intell. Syst. Technol.*, vol. 8, no. 2, oct 2016.

[16] M. Krause, C. Weiß, and M. Müller, "Soft dynamic time warping for multi-pitch estimation and beyond," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[17] J. Zeitler, S. Deniffel, M. Krause, and M. Müller, "Stabilizing training with soft dynamic time warping: A case study for pitch class estimation with weakly aligned targets," in *Proceedings of the 24th International Society for Music Information Retrieval Conference, ISMIR 2023, Milan, Italy, November 5-9, 2023*, A. Sarti, F. Antonacci, M. Sandler, P. Bestagini, S. Dixon, B. Liang, G. Richard, and J. Pauwels, Eds., 2023, pp. 433–439.

[18] F. Simonetta, S. Ntalampiras, and F. Avanzini, "Audio-to-score alignment using deep automatic music transcription," in *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*, 2021, pp. 1–6.

[19] C. Joder, S. Essid, and G. Richard, "Learning optimal features for polyphonic audio-to-score alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 10, pp. 2118–2128, 2013.

[20] Y. Wu, T. Carsault, and K. Yoshii, "Automatic chord estimation based on a frame-wise convolutional recurrent neural network with non-aligned annotations," in *27th European Signal Processing Conference, EUSIPCO 2019, A Coruña, Spain, September 2-6, 2019*. IEEE, 2019, pp. 1–5.

[21] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, ser. AMCMM '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 21–26.

[22] A. Degani, M. Dalai, R. Leonardi, and P. Migliorati, "Harmonic change detection for musical chords segmentation," in *2015 IEEE International Conference on Multimedia and Expo, ICME 2015, Turin, Italy, June 29 - July 3, 2015*. IEEE Computer Society, 2015, pp. 1–6.

[23] P. Ramoneda Franco and G. Bernardes de Almeida, "Harmonic Change Detection from Musical Audio," in *AMCMM 2006*, 2017.

[24] B. Sharma, C. Gupta, H. Li, and Y. Wang, "Automatic lyrics-to-audio alignment on polyphonic music using singing-adapted acoustic models," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 396–400.

[25] C. Gupta, R. Tong, H. Li, and Y. Wang, "Semi-supervised lyrics and solo-singing alignment," in *International Society for Music Information Retrieval Conference*, 2018.

[26] D. Stoller, S. Durand, and S. Ewert, "End-to-end lyrics alignment for polyphonic music using an audio-to-character recognition model," 2019.

[27] J. Huang, E. Benetos, and S. Ewert, "Improving lyrics alignment through joint pitch detection," 2022.

[28] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376.

[29] C.-C. Chiu, J. Qin, Y. Zhang, J. Yu, and Y. Wu, "Self-supervised learning with random-projection quantizer for speech recognition," 2022.

[30] M. Won, Y.-N. Hung, and D. Le, "A foundation model for music informatics," 2023.

[31] N. C. Tamer, Y. Özer, M. Müller, and X. Serra, "High-resolution violin transcription using weak labels," in *Ismir 2023 Hybrid Conference*, 2023.

[32] Q. T. Duong, D. H. Nguyen, B. T. Ta, N. M. Le, and V. H. Do, "Improving self-supervised audio representation based on contrastive learning with conformer encoder," in *Proceedings of the 11th International Symposium on Information and Communication Technol-*

ogy, ser. SoICT '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 270–275.

[33] Y. Chae, J. Koo, S. Lee, and K. Lee, "Exploiting time-frequency conformers for music audio enhancement," in *Proceedings of the 31st ACM International Conference on Multimedia*, ser. MM '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 2362–2370.

[34] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *Interspeech 2019*. ISCA, Sep. 2019.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[36] B. McFee and J. P. Bello, "Structured training for large-vocabulary chord recognition," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, S. J. Cunningham, Z. Duan, X. Hu, and D. Turnbull, Eds., 2017, pp. 188–194. [Online]. Available: https://ismir2017.smcnus.org/wp-content/uploads/2017/10/77_Paper.pdf

[37] L. O. Rowe and G. Tzanetakis, "Curriculum learning for imbalanced classification in large vocabulary automatic chord recognition," in *Proceedings of the 22nd International Society for Music Information Retrieval Conference, ISMIR 2021, Online, November 7-12, 2021*, J. H. Lee, A. Lerch, Z. Duan, J. Nam, P. Rao, P. van Kranenburg, and A. Srinivasamurthy, Eds., 2021, pp. 586–593. [Online]. Available: https://archives.ismir.net/ismir2021/paper/000073.pdf

[38] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler, "OMRAS2 metadata project 2009," in *International Society for Music Information Retrieval (ISMIR)*, 2009.

[39] J. A. Burgoyne, J. Wild, and I. Fujinaga, "An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis," in *International Society for Music Information Retrieval (ISMIR)*, vol. 11, 2011, pp. 633–638.

[40] H. V. Koops, B. de Haas, J. A. Burgoyne, J. Bransen, A. Kent-Muller, and A. Volk, "Annotator subjectivity in harmony annotations of popular music," *Journal of New Music Research*, vol. 48, no. 3, pp. 232–252, 2019.

[41] B. Di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro, "Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony," in *Proceedings of the 8th International Workshop on Multidimensional Systems*. VDE, 2013, pp. 1–6.

[42] A. Berenzweig, B. Logan, D. P. Ellis, and B. Whitman, "A large-scale evaluation of acoustic and subjective music-similarity measures," *Computer Music Journal*, pp. 63–76, 2004.

[43] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC Music Database: Popular, Classical and Jazz Music Databases," in *International Society for Music Information Retrieval (ISMIR)*, vol. 2, 2002, pp. 287–288.

[44] C. Weiß, F. Zalkow, V. Arifi-Müller, M. Müller, H. V. Koops, A. Volk, and H. G. Grohganz, "Schubert Winterreise dataset: A multimodal scenario for music analysis," *Journal on Computing and Cultural Heritage (JOCCH)*, vol. 14, no. 2, pp. 1–18, 2021.

[45] M. Pfleiderer, K. Frieler, J. Abeßer, W.-G. Zaddach, and B. Burkhart, Eds., *Inside the Jazzomat - New Perspectives for Jazz Research*. Schott Campus, 2017.

[46] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra, "JAAH: Audio-aligned jazz harmony dataset," Jun. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1290737

[47] M. Mauch, H. Fujihara, and M. Goto, "Lyrics-to-audio alignment and phrase-level segmentation using incomplete internet-style chord annotations," in *7th Sound and Music Computing Conference (SMC2010)*, 01 2010.

[48] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, "Lyricsynchronizer: Automatic synchronization system between musical audio signals and lyrics," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1252–1261, 2011.

[49] N. L. Masclef, A. Vaglio, and M. Moussallam, "User-centered evaluation of lyrics-to-audio alignment." in *ISMIR*, 2021, pp. 420–427.

[50] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis, "Mir_eval: A transparent implementation of common mir metrics." in *Proceedings of the 15th International Conference on Music Information Retrieval*, 2014, pp. 367–372.

[51] S. Ewert, M. Muller, and P. Grosche, "High resolution audio synchronization using chroma onset features," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 1869–1872.

[52] M. Müller, H. Mattes, and F. Kurth, "An efficient multiscale approach to audio synchronization," in *ISMIR 2006, 7th International Conference on Music Information Retrieval, Victoria, Canada, 8-12 October 2006, Proceedings*, 2006, pp. 192–197.

[53] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 569–573.

PREVALENCE OF TRESILLO RHYTHM IN CONTEMPORARY POPULAR MUSIC

Pushkar JAJORIA (pushkarjajoria@gmail.com) (0009-0006-3789-5372)[1,2],
Aurel Ruben MÄDER (aurel.mader@gmail.com) (0009-0005-9519-9896)[2], **Florian KRENN**[2], and
James MCDERMOTT (james.mcdermott@universityofgalway.ie) (0000-0002-1402-6995)[1]

[1]**Data Science Institute and School of Computer Science, University of Galway**, Galway, Ireland
[2]**Digital and Cognitive Musicology Lab, École Polytechnique Fédérale de Lausanne**, Lausanne, Switzerland

ABSTRACT

This paper presents an investigation into the prevalence and significance of the Tresillo rhythm in contemporary popular music. Initially, the Tresillo rhythm is defined and formalized within a computational framework. Subsequently, employing mathematical representations, its presence is traced within the US Billboard Top 20 Charts spanning the last two decades. Detection and quantification of the Tresillo rhythm within musical compositions are achieved through the calculation of similarities between established formalizations of the rhythm and individual songs.

The resulting similarity metrics provide insights into the degree of resemblance between a given pop song's rhythm and the predefined Tresillo pattern. To ensure the robustness of the findings, two distinct formalizations of the Tresillo rhythm are constructed, and multiple methodologies for computing rhythm similarities are systematically assessed and compared. Leveraging these similarity measures, an empirical examination of Tresillo rhythm usage in the US Billboard Top 20 Charts from 1999 to 2019 is conducted. Finally, potential explanations for observed trends are discussed and analyzed.

1. INTRODUCTION

The Tresillo rhythm, a musical motif with roots tracing back to African musical traditions, emerged as a significant cultural artifact during the era of the Atlantic Slave Trade, particularly manifesting in the Caribbean region. Notably, its distinctive cadences and syncopated beats found prominence in Cuba, where it garnered substantial recognition before crossing geographical boundaries to influence musical compositions worldwide [1, 2]. Serving not only as an independent rhythmic framework but also as an integral component within diverse musical genres, the Tresillo rhythm exemplifies remarkable versatility and adaptability [2]. As described by Floyd [2], the fundamental structure of the Tresillo rhythm is characterized by a sequence comprising a dotted eighth note succeeded by a sixteenth note, followed by an eighth rest, and culminating in another eighth note, iterated twice within a standard 4/4 musical measure. Another way to say the same thing is that the *inter-onset intervals* as defined by Toussaint are 3, 3, 2, 3, 3, 2. Slight modifications to this rhythmic template have produced other musical rhythms. By adding a beat on all fours to the rhythm, the Reggaeton rhythm is realised (see Figure 3). By changing the second half of the bar to an eighth rest, two eighth notes and another eighth rest, one creates the basic Clave rhythm pattern (see Figure 2).

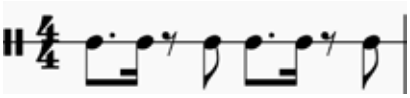


Figure 1. Synthetic Tresillo

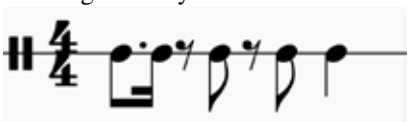


Figure 2. Clave rhythm

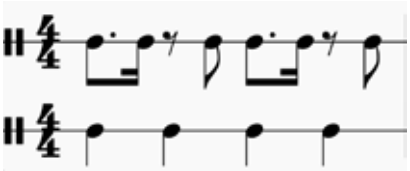


Figure 3. Reggaeton Rhythm

These subtle variations in rhythmic configuration underscore the rich dynamism and potential inherent within the Tresillo motif, illustrating its enduring relevance and widespread influence across diverse musical landscapes.

The Tresillo rhythm, notably prevalent in Latin American music, has influence beyond these origins, significantly marking its presence in Western music genres as well. This "danceable Cuban Clave son," as described by Toussaint [3], transcends geographical and cultural boundaries, finding its way into the rhythm sections of Western music compositions [4, 5]. Its popularity in contemporary pop music is evidenced by chart-topping hits such as "Shape of You" by Ed Sheeran and "Cheap Thrills" by Sia. An analysis of Billboard Chart hits, to be presented in Section 5, reveals the Tresillo rhythm's regular incorporation into pop music, underscoring its widespread appeal. While the use, evolution, and popularity of the Tresillo rhythm have been extensively explored in qualitative studies [1, 2, 4–6], a quantitative investigation into its prevalence within the

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

pop music landscape remains to be done. This gap highlights a promising area for further research, potentially offering novel insights into the rhythm's enduring popularity and its impact on global music trends.

In pursuit of a computational approach to analyze the prevalence of the Tresillo rhythm in contemporary pop music, the authors devised a novel computational method. Similar to the case of tonal key detection, we proceed by matching a histogram representing a single piece against a template histogram. Stronger matches indicate more of the Tresillo property. As in the case of tonal key detection, we can either define a binary template a priori [7], or derive a real-valued template from a corpus [8]. This method entails the digital encoding of (1) a pop song's primary rhythm as a histogram of intra-bar onset times and (2) the Tresillo rhythm itself as an analogous "binary histogram". Each of these gives a *rhythm vector*. This enables a direct comparison through various similarity metrics. Utilizing these metrics, our study reveals a discernible trend in the adoption of the Tresillo rhythm within the US Billboard Top 20 Charts over a two-decade period (1999-2019). The methodology's core is based on the premise that rhythmic structures can be quantitatively compared, thus offering a lens through which the evolution of musical tastes and influences can be traced. This computational analysis not only confirms the Tresillo rhythm's growing influence in pop music but also illustrates its role in shaping contemporary musical landscapes.

To answer the research questions, a way of defining the main rhythm of a pop song is necessary. The vast majority of pop songs consist of a simple melodic and rhythmic structure, called a *timeline* by Toussaint [3]. Following Toussaint, we therefore assume that one can identify one dominant rhythm per song. This rhythm is repeatedly played throughout the song and therefore can be characterised by counting the onsets and comparing the onset counts for every position within a bar. To present the music in a usable format, quantification is needed. 16th notes are chosen as the smallest unit. Assuming that all songs used for our analysis are in a 4/4 meter, this gives 16 possible events per bar. All songs in the data which are found not to be in 4/4 are excluded from the analysis. Aggregating all bar onsets of a song to one bar results in a single pseudo-bar which can be described as a 16-dimensional vector, where every value represents the number of onsets on a given bar position. The Tresillo rhythm is used as a rhythm on its own ("clean"), or as part of other more complex rhythms. For our definition of clean Tresillo rhythm, we use the notation in Figure 1.

This paper is organized as follows: First, the literature of several different fields, which are relevant to this paper, will be discussed. Then several assumptions necessary to conduct the presented analysis will be stated in the problem statement section. The Data section discusses the chosen data sources and data format for the analysis. In the Method section, the proposed data representation, rhythm similarity measures, and evaluation metrics are presented and explained. In the Results section, the different rhythm similarity measures are evaluated and compared. The Results section also comprises a description and analysis of the time trend. The paper concludes with a discussion of the chosen methods and obtained results. Furthermore, a possible interpretation of the produced results is presented.

2. RELATED WORK

This paper touches upon different scientific fields such as musicology, audio retrieval and digital musicology.

Prior works have already investigated the evolution and spread of syncopation in American popular music. VanderStel et. al. found that there was a general increase in syncopation over the course of the 20th century [9]. The authors focused on vocal melodies considering one song from each year of the century. Other studies focused on specifically the Tresillo and Clave rhythm patterns and thus provide a clear definition and formalization of those rhythmic patterns from a theoretical viewpoint [1, 2, 6]. We rely on those theoretical accounts to define the Tresillo rhythm used in this project. Music scholars further investigated the diffusion of the Tresillo rhythm from Africa to Latin America and then to United States from a cultural perspective [1, 2]. More generally, there have also been several works discussing the rise in popularity of Latin American music and its influence on U.S. mainstream music [10, 11]. However, the mentioned musicology research is predicated upon qualitative analyses of musicology books, sheet music, recordings and interviews with specialists and practitioners. This paper in contrast chooses to employ computational methods to draw conclusions about the influence and popularity of the Tresillo rhythm in US popular music. Another relevant research area concerns the formalization of rhythm and statistical corpus studies of rhythmic patterns. Previous authors have studied rhythm from a theoretical and generative point of view [12] and from a psychological point of view [13]. Rahn [14] analysed rhythm from a motor (physical movement) point of view [15].

Our paper mainly refers to rhythm representations, which have been used for corpus studies [16], or more generally the study of onset frequency distributions [17]. To represent rhythm in this project we will thus employ rhythm histograms as used and described in prior works [16, 17].

A last field of research that is also relevant for this paper is concerned with computing rhythmic similarity between different pieces. Such techniques are often used for audio retrieval tasks [18] or music genre classification tasks [19, 20]. More generally, this literature is concerned with measuring similarity and dissimilarity of audio signals or signals in general [21]. This literature provides valuable metrics and techniques to compare the rhythmical structure of two songs, however, it is mainly based on using audio files to extract signal features [18–20, 22]. Thus the methods proposed in those papers have been adjusted in our work to work with our symbolic data representation. In addition to this, we employ machine learning-based methods over the rhythm vectors for producing our trends.

3. DATA

In this study, four distinct data sets from different sources were utilized. To assess the effectiveness of our methodology, which is designed to calculate the similarity between the specified Tresillo rhythm and any given song, we compiled two validation data sets. The first data set comprises songs characterized by the Tresillo rhythm, while the second includes songs without this rhythm. Both data sets were carefully curated by the authors through an extensive review of songs available on Spotify to identify relevant examples.

Specifically, for songs featuring the Tresillo rhythm, we examined a pre-existing Spotify playlist purported to include such songs [1] . Following the identification of potential tracks for the validation set, we sought corresponding MIDI files on MIDIdb [2] , a MIDI file database, from which the files were subsequently downloaded. This selection process ensured that the data sets accurately represent the presence and absence of the Tresillo rhythm, thereby facilitating an evaluation of the proposed similarity measures.

To trace the Tresillo rhythm in the popular music of the past 20 years a publicly available data set which contains the song names and artist names of the Hot 100 US Billboard Charts (1999-2019) was used [3] . However to reduce the complexity of the data collection, it was decided to only use the US Billboard Top 20 Charts (1999-2019), which consists of in total of 1,447 songs. Given artist and song names, we collected the available songs of the US Billboard Top 20 Charts (1999-2019) from the website MIDIdb. The final Billboard data set on which the analysis was conducted on, consists of 444 distinct Billboard Top 20 weekly songs, which represent around 31% of the US Billboard Top 20 Charts. To assert the representativeness of the collected sample, the sample distribution was compared to the ground truth distribution of the US Billboard Top 20 Charts by evaluating t-test statistics of several features (e.g.: weeks on charts, peak position in charts, date of release). All t-tests indicate that the two distributions are not significantly different. While not conclusive, given the large sample size this may indicate that the subset is indeed representative.

MIDI files offer much broader availability and also their ability to contain multiple voices within a single song. This feature is particularly beneficial for analyzing pop songs, which often are not available in traditional score format. However, to generate onset lists for every musical event – essential for our analysis – MIDI files were converted into the Musescore format. These onset lists, detailing the timing of musical events, serve as the foundation for our data representation and subsequent analysis.

Using these onset lists, we created Figure 4, which illustrates the frequency of musical onsets, relative to the beginning of the bar, notated in 1/128 notes and aggregated into one pseudo-bar. This visual representation aids in understanding the rhythmic structure and frequency of musical events within the songs analyzed. Observing Figure 4, it becomes clear that a granularity of 16th notes is well suited.

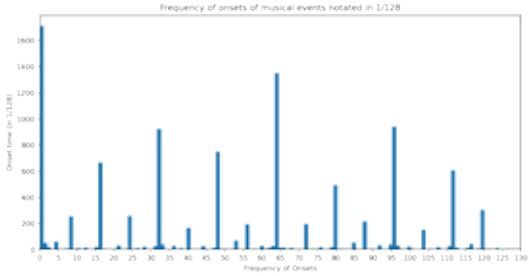


Figure 4. Frequency of onsets of musical events notated in 1/128 aggregated to one bar.

Moreover, Musescore's provision of time signatures for each song enriched our dataset with critical metadata. It is noteworthy that our dataset contained 9 songs with time signatures of 3/4 or 6/8. These songs were excluded from further analysis due to their unique rhythmic structures, which differ significantly from the common 4/4 meter found in most pop music and the focus of our study on the Tresillo rhythm.

The full dataset along with the code is available at the Github repository. [4]

4. METHODS

4.1 Rhythm vectors

To accurately measure the similarity between two rhythms, it is imperative to establish a precise definition of rhythm. Rhythm can generally be defined as "a series of onsets and durations of musical events" [12]. This study focuses on the predominant and recurring rhythm of a song, under the assumption that each musical event is adequately represented by its onset.

In the pursuit of a computational representation of a song's dominant rhythm, we employ a method where all musical onsets within a voice are aggregated into a single pseudo-bar. This approach of collapsing musical onsets into a singular pseudo-bar to generate onset 'histograms' is a well-established technique. It has been employed in various musicological studies, including analyses of Western classical music [17] and American folk music [16]. This methodological framework allows for a concise representation of rhythmic patterns, facilitating the comparative analysis of rhythm similarities across different pieces.

Given prior assessment of the Billboard data (see Figure 4) in conjunction with our choice to work only with songs in a 4/4 meter, we can safely choose to use 16-dimensional vectors for the representation of rhythm.

This method provides for each voice of each song a 16-bin histogram denoting the cumulative number of onsets on a given beat. Here it is important to note that while certain voices within a song may convey more significant informa-

[1] https://open.spotify.com/playlist/17Na5AMmlLwY7OTsw6ovsS
[2] https://www.mididb.com/
[3] https://www.kaggle.com/danield2255/data-on-songs-from-billboard-19992019
[4] github.com/pushkarjajoria/tresillo

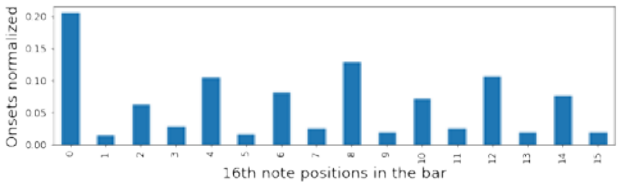


Figure 5. All rhythm vectors of all Billboard songs aggregated to one vector.

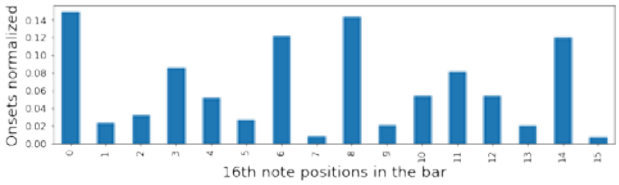


Figure 6. Rhythm vector of the song “Shape of you" by Ed Sheeran.

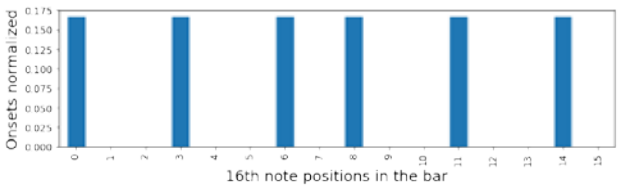


Figure 7. Rhythm vector of the synthetic Tresillo. We use this definition of the Tresillo rhythm computationally. This clean version of the Tresillo is referred to as the Synthetic Tresillo in the following sections.

tion about its main rhythm, distinguishing and quantifying the contribution of each voice to the overall perception of the rhythm requires specialized knowledge. Developing a methodology to ascertain the extent to which each voice influences the perception of the main rhythm falls outside the ambit of this study. Therefore, this paper does not differentiate between the voices, treating them uniformly in the analysis. By aggregating the onsets across all voices onto one single histogram, we obtain a single onset histogram for a given song. The onset histograms are then normalized to transform them into a 16-dimensional vector, which we will refer to as the rhythm vector of a song. This research assumes that the information about the rhythm of the song is captured by these rhythm vectors.

The obtained rhythm vectors can be displayed as bar plots to allow visual inspection. Aggregating all rhythms vectors into one normalized rhythm vector shows the mean rhythm of our Billboard data set, as can be seen in Figure 5. Figure 6 shows a song with high Tresillo similarity.

Next, we introduce in Figure 7 the synthetic Tresillo pattern. Visual inspection and comparison of the compiled rhythm histograms to the synthetic Tresillo pattern reveals similarities and differences. Songs that contain the Tresillo rhythm show a strong visual similarity with the synthetic Tresillo rhythm vector.

4.2 Tresillo similarity measures

Each rhythm vector is a 16-dimensional vector for which similarity compared to another vector in the same space can be computed using a cosine similarity measure. We compare the similarity of rhythm vectors with two different Tresillo vectors which are defined as follows. 1) Template similarity center is the point in this space that corresponds to a plain Tresillo beat. 2) Centroid similarity point is defined as the centroid of all rhythm vectors corresponding to Tresillo songs under the training dataset. Our purpose in defining both methods is to explore and experiment to find the best method.

The Tresillo rhythm is defined by its syncopated pattern. This information is visible in Figure 6 by a sharp peak on the 3rd and 12th beats. Since each rhythm is defined by a pattern of higher and lower values along these 16 dimensions, it is fair to assume that each axis or onset position does not carry equal weight in the identification of this pattern. For example, it is very common in songs to have an onset at the start of a bar. Since the onset on the first beat is so ubiquitous in music, it need not carry a high weight in the identification of a rhythm in this space. To encapsulate this information in a similarity measure, we next learn scaling factors for each dimension of the rhythm space, which are referred to as θ_i. These θ_i are used to increase the gap between “Tresillo similarities” of rhythm vectors which do contain Tresillo and vectors which don’t. These θ_i scale the rhythm vector of a song along axis i and in turn scale the similarity measure accordingly. The resulting parameterized cosine similarity is defined in equation 1 where, Θ refers to the set of scaling factors, and A and B are two vectors with the same dimension between which the similarity needs to be computed. The ith dimension of these vectors are denoted by a_i and b_i. A_Θ and B_Θ are the linearly transformed vector after scaling i^{th} dimension by θ_i. *‘n’* denotes the total number of dimensions in the rhythm space, i.e. 16 in our case. Parameterized distance measures have been successfully used in the past in pattern recognition and machine learning [21].

$$\begin{aligned} \cos_\Theta(\mathbf{A},\mathbf{B}) &= \frac{\mathbf{A}_\Theta \cdot \mathbf{B}_\Theta}{||\mathbf{A}_\Theta||\,||\mathbf{B}_\Theta||} \\ &= \frac{\sum_{i=1}^{n}(\mathbf{a}_i \cdot \theta_i)(\mathbf{b}_i \cdot \theta_i)}{\sqrt{\sum_{i=1}^{n}(\mathbf{a}_i \cdot \theta_i)^2}\sqrt{\sum_{i=1}^{n}(\mathbf{b}_i \cdot \theta_i)^2}} \end{aligned} \tag{1}$$

Equation 1 defines the parameterized cosine similarity used in this research. The parameters for this model are learned by maximizing S^* (defined in the next section), which can also be modeled as a minimization problem as formulated in Equation 2.

$$\underset{\Theta}{\operatorname{argmin}} \frac{cos_\Theta(\mathcal{A}',\mathcal{T})}{cos_\Theta(\mathcal{A},\mathcal{T})} \tag{2}$$

Where $\mathcal{A}$ is the set of songs with Tresillo present in them, $\mathcal{A}'$ is the set of songs with Tresillo not present in them and $\mathcal{T}$ refers to the reference point for computing the cosine similarity, i.e. either the clean Tresillo rhythm template, or the Tresillo centroid vector.

4.3 Evaluation

To evaluate the proposed similarity methods two different metrics were chosen to assess the variance produced by a

given model and to compare the model fits between different models.

To assess the variance in Tresillo similarity estimated by a given model, the bootstrapping method on the validation data sets was used. Thus using the Tresillo and the non-Tresillo validation data sets, we used a given model to calculated the mean Tresillo similarity in a given validation set and its 95% confidence intervals, as obtained by bootstrapping. The bootstrapping was performed with 1000 draws with replacement. The number of samples per draw, correspond to the sample size of a given validation set (i.e either 9 or 10 samples).

To compare different models it was assumed that a good model would have high similarity for all songs which have a Tresillo pattern and a low similarity for songs which do not have such a pattern (see Equation 3). This can be measured by defining 'Similarity Goodness' S^* as the ratio of mean similarity in songs that have Tresillo and mean similarity of songs that do not. A higher value of S^* denotes high similarity for songs with Tresillo and low similarity for songs without Tresillo. Here similarity refers to the similarity computed between the rhythm vector of a song and a Tresillo rhythm vector. This similarity could be computed by using either an unparameterized or a parameterized cosine similarity, depending on the different models defined in the previous section.

$$S^* = \frac{S_{\text{tres}}}{S_{\text{no-tres}}} \tag{3}$$

In Eq. 3, S^* denotes the 'Similarity goodness', a ratio indicating the relative similarity among songs with and without the Tresillo rhythm. Specifically, S_{tres} represents the average similarity score for songs incorporating the Tresillo rhythm, calculated based on the cosine similarity between their rhythm vectors and a defined Tresillo rhythm vector. On the other hand, $S_{no-tres}$ denotes the average similarity score for songs that do not feature the Tresillo rhythm. It is extremely unlikely to get zero in the denominator as the tresillo vector and the song vector would need to be orthogonal, i.e. having zero notes on the first beat of the bar is unlikely. A higher value of S^* suggests that songs with the Tresillo rhythm are, on average, more similar to the Tresillo rhythm vector compared to songs without it, highlighting the distinctive rhythmic pattern associated with the Tresillo rhythm.

5. RESULTS

5.1 Comparing the similarity measures

Figure 8 compares $\mathcal{S}*$ using two different Tresillo centers with a parameterized vs unparameterized cosine similarity. The error lines on the bar plot denote the 97.5% confidence interval based on 'leave one out' cross-validation. It can be inferred that the models based on the synthetically defined Tresillo outperform the models based on the centroid methods ($p < 0.001$ in both cases using a t-test). Parameterized models also outperform the un-parameterized models (again, $p < 0.001$ in both cases using a t-test).

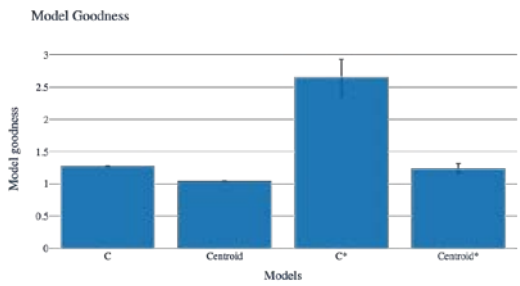


Figure 8. Comparing model goodness. Here, **C** refers to rhythm similarity measured with cosine similarity, using Tresillo template as centre. **Centroid** refers to rhythm similarity measured with cosine similarity, using the centroid of Tresillo songs as centre. **C*** refers to rhythm similarity measured with parameterized cosine similarity, using Tresillo template as centre. **Centroid*** refers to rhythm similarity measured with parameterized cosine similarity, using the centroid of Tresillo songs as centre.

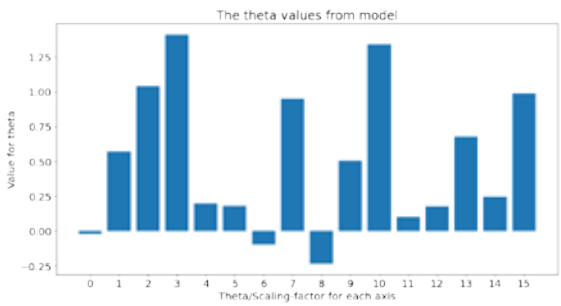


Figure 9. θ_i for each dimension i (i.e. 16th beat).

Figure 9 shows the learned θ_i for each beat after fitting the model. High values here indicate that this location is informative as to *whether* the beat is a Tresillo or not. Low and negative values for the 0th and 8th beats bolster our claim about the ubiquitous 0th beat in popular music. The third beat along with the second beat carry a lot of information about the Tresillo beat and hence have a high value. Other Tresillo beats also share a high peak with the exception of the 6th and the 8th beat. This may be because the onsets for popular rock pop songs coincide here. There is asymmetry across the 8th beat. One of the possible reason for this could be the drum fills, which are often in the latter half of the bar.

5.2 Trend over Time

Considering the evaluation of the proposed models based on our validation data sets, it was inferred that models using the synthetic Tresillo outperformed models using the centroid derived from data. Thus to analyze the Tresillo trend over time, only the two models which are based on the synthetic Tresillo were used. Those two models correspond to the models with best performance on the validation set. However, the parametrized model clearly outperformed the cosine similarity model.

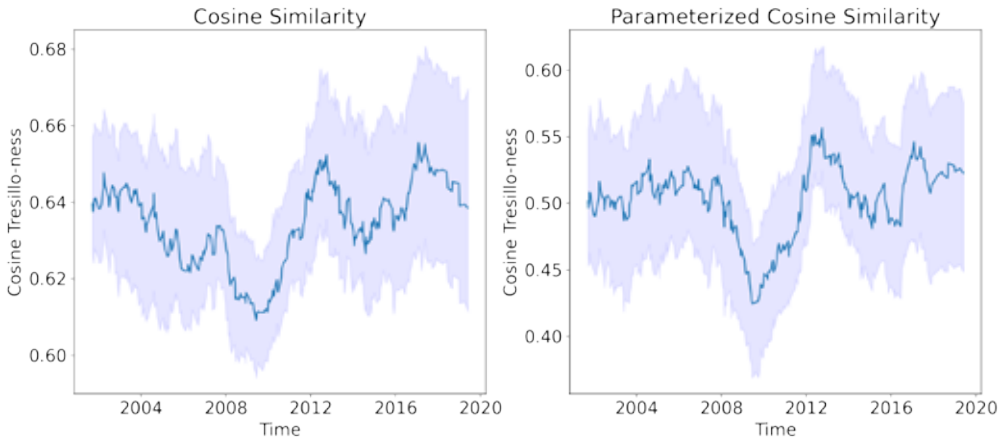


Figure 10. 52 weeks moving average of cosine Tresillo similarity in the US Billboard Top 20 Charts.

Tresillo Songs Names	$\mathcal{C}$	$\mathcal{C}^*$
Cheap Thrills-Sia	0.9089	0.9919
Eastside-benny blanco	0.7112	0.9471
New rules-Dua Lipa	0.8388	0.9431
Non Tresillo Songs Names	$\mathcal{C}$	$\mathcal{C}^*$
Immigrant Song	0.5842	0.5382
Fear of the Dark	0.6894	0.2545
Sweet Child O'Mine	0.7247	0.4840

Table 1. Cosine similarity and parameterized cosine similarity for popular Tresillo songs and Non Tresillo. Here $\mathcal{C}$ denotes cosine similarity and $\mathcal{C}^*$ denotes parameterized cosine similarity.

Given those two models, Tresillo similarity for all songs in the Billboard data set was calculated. By plotting the Tresillo similarity over time, a proxy for the trend in Tresillo use over time was obtained. A rolling yearly mean was applied to the weekly mean value in Tresillo use. The resulting rolling yearly average of Tresillo use can be seen in Figure 10. In both Figures, 95% confidence intervals have been obtained via bootstrapping (1,000 draws with replacement, with the number of samples per draw equal to the sample size) and indicated by light blue coloring.

6. DISCUSSION

This paper shows that, given a clear definition, the intensity of Tresillo rhythm use in a given song can be measured with computational methods. Several methods have been introduced which identify the use of the Tresillo rhythm and its intensity in the collected validation data sets. However, it is still disputable how the proposed models deal with noise which might dilute the Tresillo rhythm. Assessing the uncertainty of our models (e.g.: by looking at outliers as determined by the bootstrapping method), it is notable that not every song which was labeled to contain the Tresillo rhythm, has a very high Tresillo similarity. Although both models in Table 1 give high similarity for a Tresillo song and low similarity for a 'non-Tresillo' song, parameterized model $\mathcal{C}^*$ gives the desired larger gap between the two.

Assessing the trends over time of Tresillo use in the US Billboard Top 20 Charts of the past 20 years, no clear linear trend is observable. However, several interesting peaks and patterns are noticeable. We also found that the results at a granularity of a week are by nature very noisy and there is high variance in Tresillo use from week to week, which is expected and does not offer any meaningful insight.

Using a rolling yearly mean, the trend of Tresillo use over time becomes more visible, as can be seen in Figure 10. Figure 10 offers us some interesting insights. Even though there is considerable yearly variance in the intensity of Tresillo rhythm use (as illustrated by the 95% confidence intervals), there are identifiable peaks and valleys in Tresillo rhythm use. Furthermore, the unparameterised cosine similarity and the parametrized cosine similarity produce consistent results, however on different scales.

Looking at Figure 10, we observe a trend which starts relatively high around the new millennium, stays constant or slightly decreases till around 2008, and collapses to a low in 2010. From 2010 on there is an increase in Tresillo use, with another valley around 2014. Around 2018 the trend in Tresillo rhythm intensity reaches its all time high.

After subjectively evaluating the calculated Tresillo similarity and the corresponding Billboard Charts, we interpret this trend as follows. In the early 2000s it seems that many songs which peaked in the Billboard Charts were either by Latin artists or by artists who used Latin music themes in their songs (e.g.: *Maria Maria*, Santana, 2000; *Be With You*, Enrique Iglesias 2000; *Baby Boy*, Beyoncé, 2003). This trend then decreases steadily, till the Tresillo rhythm seems to reappear in Western dance music after 2010. After 2010 there are several peaks which can be associated to popular dance music songs with particularly high Tresillo similarity (e.g.: *Where Have You Been*, Rihanna, 2012; *Shape Of You*, Ed Sheeran, 2017; *Cheap Thrills*, Sia, 2016). The trend post 2010 agrees with the findings of VanderStel and Temperley [9]. Despite the valleys of 2010 observed in our findings, it is our opinion that the overall trend exhibits an upward trajectory, albeit

a modest one. However, to substantiate this interpretation of the trend over time, further empirical research would be needed, which clearly defines and differentiates the use of the Tresillo rhythm in the context of 'Latin American' music and its usage in Western dance music.

7. FUTURE WORK

The channels in a song carry valuable information and could be leveraged upon if a sophisticated algorithm could be developed which is independent to meta data information but rather works on a symbolic level.

The lack of well annotated midi data is also a limiting factor. Annotating more data will result in better parameterized models which in turn, would improve the $\mathcal{S}^*$. The benefits of more data are not limited to this. More sophisticated learning algorithms which could not be used given over-fitting concerns, might become viable. For instance, a non linear transformation of the rhythm vector space may results in better results as this would be better suited at modeling the nuances on, for instance the 3rd beat. Dimension reduction techniques like PCA could also be employed to reduce over-fitting.

Finally, to substantiate our subjective impression that there are two waves in usage of this rhythm, once in Latin American music and once in Western dance music, further empirical research would be needed, which differentiates in which musical context this rhythm is used.

Acknowledgments

This work was carried out at the Digital and Cognitive Musicology Lab at EPFL and at the University of Galway with funding from the Polifonia Project of the European Union's Horizon 2020 research and innovation programme under grant agreement N. 101004746 and with the financial support of Science Foundation Ireland under Grant Number 12/RC/2289_P2 - Insight SFI Centre for Data Analytics. For the purpose of Open Access, the author has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising from this submission.

8. REFERENCES

[1] A. Acquista, *Tresillo: A rhythmic framework connecting differing rhythmic styles*. California State University, Long Beach, 2009.

[2] S. A. Floyd, "Black music in the circum-Caribbean," *American Music*, pp. 1–38, 1999.

[3] G. Toussaint, "The geometry of musical rhythm," in *Japanese Conference on Discrete and Computational Geometry*. Springer, 2004, pp. 198–212.

[4] W. A. Sethares, "The geometry of musical rhythm: what makes a "good" rhythm good?" 2014.

[5] N. Biamonte, "Rhythmic functions in pop-rock music," in *The Routledge Companion to Popular Music Analysis*. Routledge, 2018, pp. 190–206.

[6] C. Sandroni, *Le Tresillo rythme et «métissage» dans la musique populaire Latino-Américaine imprimée au XIXe siècle*. Infolio Editeur/Ateliers d'ethnomusicologie, 2001.

[7] H. Bellmann, "About the determination of key of a musical excerpt," in *International Symposium on Computer Music Modeling and Retrieval*. Springer, 2005, pp. 76–91.

[8] C. L. Krumhansl and E. J. Kessler, "Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys." *Psychological review*, vol. 89, no. 4, p. 334, 1982.

[9] J. VanderStel and D. Temperley, "The evolution of syncopation in twentieth-century american popular music," *Journal of New Music Research*, vol. 51, no. 2-3, pp. 162–185, 2022. [Online]. Available: https://doi.org/10.1080/09298215.2023.2223583

[10] J. S. Roberts, "The Latin tinge: the impact of Latin American music on the United States," *Oxford University Press*, 1999.

[11] P. T. Schroeder, "The growth of Latin American pop music in the United States," in *College Music Symposium*, vol. 18. JSTOR, 1978, pp. 124–129.

[12] M. Rohrmeier, "Towards a formalization of musical rhythm," in *Proc. of the 21st Int. Society for Music Information Retrieval Conf*, 2020.

[13] J. London, *Hearing in time: Psychological aspects of musical meter*. Oxford University Press, 2012.

[14] J. Rahn, "Turning the analysis around: Africa-derived rhythms and Europe-derived music theory," *Black Music Research Journal*, pp. 71–89, 1996.

[15] F. Gomez-Martin, P. Taslakian, and G. T. Toussaint, "Evenness preserving operations on musical rhythms," in *Proceedings of the 2008 C3S2E conference*, 2008, pp. 121–123.

[16] D. Huron and A. Ommen, "An empirical study of syncopation in American popular music, 1890–1939," *Music Theory Spectrum*, vol. 28, no. 2, pp. 211–231, 2006.

[17] C. Palmer and C. L. Krumhansl, "Mental representations for musical meter," *Journal of Experimental Psychology: Human Perception and Performance*, vol. 16, no. 4, p. 728, 1990.

[18] J. Foote, M. Cooper, and U. Nam, "Audio retrieval by rhythmic similarity." in *ISMIR*. Citeseer, 2002.

[19] S. Dixon, F. Gouyon, G. Widmer *et al.*, "Towards characterisation of music via rhythmic patterns." in *ISMIR*, 2004.

[20] G. Peeters, "Rhythm classification using spectral rhythm patterns." in *ISMIR*, 2005, pp. 644–647.

[21] Z. Zhang, J. Kwok, and D.-Y. Yeung, "Parametric distance metric learning with label information," *Proc International Joint Conference on Artificial Intelligence*, 05 2003.

[22] M. Panteli, N. Bogaards, A. K. Honingh *et al.*, "Modeling rhythm similarity for electronic dance music." in *ISMIR*, 2014, pp. 537–542.

PROGRAM NOTES

CONCERT 4

/ Saccades

/ Nan

/ Kurivari-live

SACCADES

Ted Moore
Composer
Jorge Sousa, saxophone
University of Aveiro, INET-md

A "saccade" is a rapid movement of the eyeball between two fixed focal points. During this brief moment, the brain hides this blurry motion from our perception. Once a saccade motion has begun, the destination cannot change, meaning that if the target of focus disappears the viewer won't know until the saccade completes. If the field of vision is changing too quickly, the saccades may never be able to arrive at and focus on a target, instead, the objects in view are only perceived through peripheral vision.

This phenomenon is imitated by the sound and video presented in the piece. It also serves as a metaphor for the density of information and high entropy experiences we're constantly trying to cope with. A scroll on social media, smartphone alerts, big data, technological advancements and predictions, the abundance of choices in the grocery aisle.

The processes used to create saccades embody this density of information–and the constant struggle to track it all. Large datasets of audio analyses derived from the tape part are sent to neural networks and dimensionality reduction algorithms to find patterns and then visualize and sonify what is found. A plethora of variations on visual themes are created by the combinatorics of stochastically triggered visual synthesis modules and processing effects. Computer vision analysis adds layers to our visual and aural perception–tightly binding together the visual and auditory elements. In the final movement, source video of an eyeball is morphed via a convolutional neural network creating eyeballs that feel both simple and obvious, but also slightly make my stomach squirm.

NAN

Ted Moore
Composer

"nan" is an audio-visual composition based on the timbre and rhythmic gating of the NAND- gate feedback circuit described by Nick Collins in "Handmade Electronic Music". Even though the system is quite simple (producing repeating phrases consisting of square waves, filtered noise, and silence), each gesture has microvariations that increase the entropy and attract my attention endlessly. My favorite timbres from this circuit occur while parameters are being changed–when capacitors are firing at surprising times, before they can settle into a stasis.

Large datasets of audio analyses derived from the tape part are sent to neural networks and dimensionality reduction algorithms to find patterns and then visualize and sonify what is found. A plethora of variations on visual themes are created by the combinatorics of stochastically triggered visual synthesis modules and processing effects. Computer vision analysis adds layers to our visual and aural perception–tightly binding together the visual and auditory elements.

KURIVARI-LIVE

Ted Moore
Composer
Jorge Sousa, saxophone
University of Aveiro, INET-md

This is an algorave project kurivari that derives its musical style from analysis of traditional rhythm patterns fused with sonic material generated with genetic sound synthesis algorithms. The rhythmic structure of the performance is built from patterns that have been stripped down to their most essential skeletal structures and are then juxtaposed during performances to create long complex cycles and unpredictable syncopation while still maintaining a clearly discernible downbeat in order to undermine the dominance 4/4 time signature of electronic dance music. The outcome is more reminiscent of rhythmic structures found in more recent subgenres of hip hop music such as drill or trap rather than the regularity of a techno beat.
Apart from a selection of samples, the sonic palette is predominantly created with [gene experssion synthesis] (https:// geen.tehis.net/), a generative synthesis technique that harnesses the power of genetic algorithms to write computer code. Gene expression synthesis combines ideas from the two most widely used and well-known evolutionary programming methods: genetic algorithms and genetic programming. The algorithm generates computer code, SuperCollider SynthDef objects in this case, by initiating a population of candidate solutions - either from a predefined selection or randomly generated - and then applies genetic operators to the population. Each candidate is defined as a list of either unit generator functions or function parameter placeholders designated with lower case letters. The translation from genotype to phenotype follows a breadth-first recursive principle: as the codons of a gene are traversed, for each function encountered, the algorithm reserves a number of following unreserved codons as arguments to that function regardless whether they are functions or terminals. Once decoded as an executable program, each candidate solution in a population is subjected to a fitness evaluation that determines how well it performs at solving a target problem, which is a similarity measure to an existing sound sample.
This method has generated thousands of SynthDef objects that are saved in a NoSQL cloud database [MongoDB] (https://www.mongodb.com/) and can be accessed in real time during a performance. Each object is also analysed with music information retrieval algorithms and the results stored in a database to facilitate easier search functionality. The audio features that are used to find synthesisers include tempo measures such as beats-per-minute (BPM), beat timings and onset rate, and spectral characteristics such as mel-bands, spectral flatness, and spectral centroid, extracted with [madmom] (https://madmom.readthedocs.io/en/latest/) and [librosa] (https://librosa.org/doc/latest/index.html) audio analysis toolkits. Beat timings also assist in identifying interesting sonic loops that can be used in the live composition process and inserted into the rhythm patterns.
Real time sound spatialisation with the [Ambisonic Toolkit] (https://www.ambisonictoolkit.net/) is another characteristic of this performance to depart from the traditional directionality of live performances. kurivari has performedonmultichannelsystemsconsistingofupto26speakersdistributedthroughoutaperformance space,but also has been easily adapted to stereo systems using binaural rendering principles.
The fusion of traditional rhythm patterns and complex modulation synthesis is intended to create a sonic environment that explores the uncertain area between rigid rhythmic structures and completely form-free noise improvisations, the edge where order and chaos meet and create interesting moments of complex- ity intermittent with simplicity. The sonic environment is complemented by 3–dimensional generative computer graphics that are based on artificial life algo- rithms and incorporate the computer code evaluated during the performance to look aesthetically more engaging than simply displaying the livecoder's screen.

OTHER PAPERS / ONLINE ONLY

/ Design and Evaluation of Accessible Digital Musical Instruments for Pupils with Neurodevelopmental Disorders

/ Biosensors in Immersive Music Performance: Andromeda for Flute and Biofeedback

/ Xtrack: Automatic Segmentation of Live Music with YAMNet

/ Cyborgutt: Exploring the body's fluid boundaries through a biosensor composition

/ Trajectory Descriptors: an Approach to Music Genre Classification through Curves and Vectors in the Tonnetz

DESIGN AND EVALUATION OF ACCESSIBLE DIGITAL MUSICAL INSTRUMENTS FOR PUPILS WITH NEURODEVELOPMENTAL DISORDERS

Matteo OLIVO (matteo.olivo@univ-st-etienne.fr) [1,2],
Florence CARROUEL (florence.carrouel@univ-lyon1.fr) [2],
Emily DARLINGTON (emily.darlington@univ-lyon1.fr) [2], and
Laurent POTTIER (laurent.pottier@univ-st-etienne.fr) [1]

[1] *Laboratory ECLLA*, **Jean Monnet University**, France
[2] *Laboratory Health Systemic Process (P2S) UR4129*, **University Claude Bernard Lyon 1**, France

ABSTRACT

Although considerations have been proposed by some researchers, little is still known on the design and the evaluation of Accessible digital musical instruments (ADMIs) for pupils with neurodevelopmental disorders (NDDs), and more generally for pupils with special educational needs (SEN). This article reports on the design process and evaluation of a series of ADMIs, specifically developed for pupils with NDDs. The series consists of 5 web applications (a Max/MSP version is also available for some of them) aimed to play a musical composition evoking a natural soundscape. Through a participatory and iterative design process, some modifications to the application interfaces were made, to cope with the users' impairments. The evaluation was carried out through a qualitative approach in a special educational class of a French middle school: 4 pupils with NDDs took part in some musical sessions aimed to test the applications, and in interviews. A special education teacher and a teaching assistant were also involved in the musical sessions and in interviews. Some considerations emerged from the evaluation that will guide the future steps of the design process.

1. INTRODUCTION

According to Frid [1], Accessible digital musical instruments (ADMIs) are "accessible musical control interfaces used in electronic music, inclusive music practice and music therapy settings". These digital instruments are based on different types of control interfaces, such as touchscreen controllers, gaze controllers or mouth-operated interfaces. This variety of devices allows to adapt the instruments to the needs of individuals with disabilities, making music more accessible. The development of ADMIs is grounded on the theory of the Social Model of Disability [2], which argued that people with disabilities are not disabled by their impairments, but by the disabling factors they face in society [3]. According to this theory, it is necessary to implement assistive technologies to overcome the barriers that disabled people face in music-making [4], and to foster their empowerment process [1].

The design method and the evaluation [5] are key aspects to be considered in the development of ADMIs [6]. Some attempts were made to provide design considerations [7, 8] and evaluation frameworks for ADMIs, based on different uses cases [9, 10]. However, there are still no established and commonly accepted design and evaluation frameworks for this kind of musical instruments [6, 11].

With regards to ADMIs design, the literature suggests that participatory method is essential [12]. The review carried out by Frid [1] showed that most of the ADMIs are developed following participatory and iterative design methods. Participatory design involves all users in the problem-solving process, allowing an understanding of holistic features of the person-user's interests, needs and preferences [13]. Participatory design takes inspiration from the motto of the disability rights movement "Nothing about us without us" [14]. This means that the design process should ensure that users are in control of the technology, and not just passive recipients of it [15]. Iterative design is a method for developing user interfaces by refining them iteratively, over several versions. Each iteration is subjected to an evaluation, through user testing or other methods [16], to determine whether or not the iteration achieved its goals [17].

Concerning the evaluation of ADMIs, some researchers [18, 19] stressed that it cannot be based on frameworks borrowed from Human-Computer Interaction, such as the task-based framework proposed by Wanderley and Orio [20]. Various authors highlighted that the evaluation of Digital Musical Instruments (DMIs) should take in consideration the sociocultural context in which musical interactions take place [6, 18, 21]; Jack et al. suggested that the evaluation of DMIs should be based on a qualitative and reflective process [21].

School music programs can play an important role for the empowerment process of pupils with neurodevelopmental disorders (NDDs). NDDs are intellectual and cognitive disorders that have an onset in the developmental period. They affect a large percentage of the school-age population [1] and include intellectual disability, autism spec-

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

[1] According to the United States Environmental Protection Agency, NDDs affect the 15% of children between 3 and 17 years old (see https://www.epa.gov/).

trum disorder, attention-deficit/hyperactivity disorder, neurodevelopmental motor disorders (dyspraxia, stereotypic movement) and specific learning disorders (dyslexia, dysgraphia, dyscalculia) [22]. There is growing research interest in the impact of school-based music activities on the wellbeing of children with NDDs [23, 24]. However, little is still known on the actual use of ADMIs in school settings by pupils with NDDs (and, more generally, by pupils with SEN), and well-grounded design considerations are rare [11]. The last review on the use of music technologies in SEN schools was published in 2011 [25]. More recent publications provided considerations regarding the design of ADMIs in SEN settings: Förster [11] highlighted the importance of including students and SEN teachers in the design process. Frid et al. [15] also stressed the need to involve different stakeholders in ADMIs design, such as parents and SEN teachers. Förster et al. [26] and Ward and Davis [7] highlighted the importance of considering the context (the school environment).

This article reports on the participatory and iterative design process and evaluation of some ADMIs, aimed at middle school pupils with NDDs. The evaluation led to some considerations that will guide the next iterations of the design process, and may benefit further research in this field.

2. THE ACCESSIBLE MUSICAL APPLICATIONS FOR PUPILS WITH NDDS

2.1 Educational Goals and Technical Features

The ADMIs described and evaluated in this study are 5 musical applications [2]. They are part of a set of 8 applications, developed by the first author (MO) for his PhD research. These ADMIs are designed for a school-based training program, based on writing and performing a musical composition, aimed to develop life skills (communication, teamworking, empathy, etc.) [27] of pupils with NDDs. The composition has to evoke a natural soundscape; it is written on the basis of a story invented by the pupils, which is then sonified using the applications. For this reason, the applications simulate sounds of a natural soundscape, or musical instruments (such as the Digeridoo or the Djembé) that are reminiscent of a natural environment. The aim of these ADMIs is to provide pupils with accessible and intuitive instruments that they can quickly play, without previous musical training. In this way, it is possible to quickly start the collective work that facilitates the life skills development. For each application a web version exists, developed in Javascript using the Web Audio API [3]; some customized web audio nodes (AudioWorkletNodes) are developed in Faust [28]. For some applications a second version is available, implemented in Max/MSP. Below we provide a description of the 5 applications (for more technical details, please refer to [29]):

- Soundscape: web application that allows the user to play recorded sounds from a natural soundscape (birds, rain, etc.), by clicking on buttons. Some sliders allow the user to add reverb, delay and a low-pass filter to the sounds. A Max/MSP version of the application is also available, which can be used with the AKAI MIDImix controller: the line faders of the controller trigger the sounds and the knobs adjust their volumes, the amount of effects and the cut-off frequency of the filters.

- The Sea: web application that simulates the sounds of the sea and waves through subtractive synthesis techniques. On smartphones and tablets, the sound of the waves can be triggered by clicking on a button or by rotating the device; this system is based on the mapping of the accelerometer, developed with a Javascript Sensor API [4].

- Didgeridoo: web application for smartphone or tablet that simulates the sound of a didgeridoo through additive synthesis techniques. The movement of the device allows to adjust the angle of a Wah pedal applied to the sound; this system, based on accelerometer mapping, simulates the sound modulations that the musician produces through the vibrations of the lips when blowing into the didgeridoo.

- Wind: web application for smartphone or tablet that simulates the sound of the wind using subtractive synthesis techniques. The mapping of the accelerometer allows to modulate the wind force, according to the movement of the device.

- Djembé: web application that simulates the sounds of the djembe (tonic, slap and bass). A 2-bar sequencer, based on buttons to press, allows the creation of basic rhythms; a slider adjusts the bpm of the sound. A Max/MSP version is also available, that can be used with the AKAI pro MPD 2018 Midi controller: the controller pads trigger the sounds of the djembé and a knob adjusts their volume.

The Midi controllers to be used with Soundscape and Djembé were chosen because they are intuitive and easy to carry, thus suitable for SEN settings.

2.2 Design Process

The design of the applications described in section 2.1 is based on a participatory and iterative process, carried out in the context of the MO's PhD. In 2021 some early-stage prototypes were realized. They were mainly developed in Max/MSP, with only some simplified web versions. Their design process was based on the analysis of the literature on ADMIs, and in particular on the works of Ward and Davis [7] and Graham-Knight and Tzanetakis [8], which provided some key-considerations: the importance of applying the participatory method and of providing feedback of the interaction; the importance of the instrument's robustness and sound quality; the fact that ADMIs should be small, cheap, easy to use and should produce sound quickly. Between February and July 2021 the applications were tested by a group of 4 adolescents with intellectual disability (Williams and Beuren syndrome) and a group of 3 adults with cognitive impairments, outside the school context: 4 musical sessions were carried out with the fist group and 5 with the second. These tests allowed a preliminary evaluation of the accessibility of the applications and their impact on music composition. The results of this evaluation are presented in this article [29]. In 2022/2023 some prototypes of the applications were further improved,

[2] matteoolivo.com
[3] https://www.w3.org/TR/webaudio/
[4] https://www.w3.org/TR/generic-sensor/

Figure 1. Example of web application interface (Djembé).

Figure 2. Interface of the application and text explaining the operation mode (The Sea).

and others were abandoned. 8 web applications prototypes are currently available; for 2 of them the Max/MSP version has been developed. The 5 applications evaluated in this study have been chosen because they are considered to be the most relevant for the composition work; however, some considerations that emerge from this study are equally applicable to the other applications.

In 2022 it was decided that the musical training program (see section 2.1) should be focused on school settings, as schools play a major role in life skills development [30]. This led to a shift in focus from cognitive and intellectual disabilities in general, to NDDs: indeed, NDDs are cognitive and intellectual disorders (listed in the Introduction) that emerge in the developmental period, affecting a significant proportion of the school-age population. Some adaptations (see Figure 1) have been made to the applications to make them more accessible to pupils with NDDs, compared to other soundscape web apps [5] or educational softwares [31], and to help pupils use these ADMIs autonomously. The adaptations are based on the tests conducted in 2021, guidelines on accessibility [32] and informal conversations carried out in 2022 with a SEN teacher:

- adaptations to cope with impairments in movement coordination (dyspraxia): the Max/MSP version of the applications can be handled by MIDI controllers. In Didgeridoo, Wind and The Sea, the triggering of sounds or the modification of the sound parameters can be controlled by the device movement. The responsive interfaces of the web applications automatically adapt to all screen sizes. This variety of options allows the users to choose the application, the device (MIDI controller, smartphone, tablet, computer) and the interaction mode (mouse, rotation, MIDI controller, touchscreen) that best suit their motor coordination skills.

- Adaptations to facilitate the comprehension of the operation mode (for web versions), for children with intellectual disability: the functionalities of the applications have been simplified, and the number of UI elements has been reduced as much as possible. The button in the top right-hand corner of the interfaces allows to access a downloadable file, containing simplified instructions on how to use the applications (see Figure 2). Labels and small icons explain the functions of the UI elements. When buttons are pressed, their colour changes to provide visual feedback of the interaction.

- Adaptations for children with dyslexia or other reading impairments (for web versions): the text on the interfaces and the instructions (see Figure 2) are written in the OpenDyslexic font [6], specifically designed for dyslexic people. Other adaptations are made to the texts (font size, line spacing, colour contrast, separation of syllables), to make them easier to read. By clicking on a button in the top left-hand corner of the interfaces, a voice synthesis system (developed using the Javascript Web speech API [7]) allows the user to listen to the instructions on the operation mode of the applications. The voice reading speed can be adjusted according to users' needs.

3. EVALUATION OF THE APPLICATIONS

3.1 Study Design and Setting

The evaluation of the 5 musical applications described in section 2.1 was based on a qualitative method, and was carried out by MO in the Waldeck-Rousseau middle school of Firminy (France): between November and December 2023, 4 8th grade middle school pupils with NDDs took part in 4 one-hour musical sessions aimed to test the applications. The pupils attended the ULIS (Unitée Localisée pour l'Inclusion Scolaire) class, a specific class for children with SEN.

3.2 Participants

4 pupils participated in the study (A, B, C, D). The inclusion criteria for the pupils were: (i) 8th grade middle school pupils, (ii) with NDDs, (iii) attending the ULIS class, (iv) who agreed to participate in the study by informed consent, (v) whose parents provided informed consent to their involvement in the study. The 4 pupils participated in the musical sessions and in interviews. They were affected by the following NDDs:

[5] See an example here: https://soundscape.world/.

[6] https://opendyslexic.org/

[7] https://wicg.github.io/speech-api/

- A: autism and attention-deficit/hyperactivity disorders.
- B: moderate intellectual disability, dyslexia, memory and communication impairments.
- C: dyslexia.
- D: moderate intellectual disability, dyslexia.

Also the teacher and the teaching assistant [33] in charge of the ULIS class (henceforth referred to as "the professionals") took part in the musical sessions and in an interview.

MO conducted the evaluation. He was a PhD student at the Jean Monnet University of Saint-Etienne, with a background and professional experience in the fields of disability, assistive technologies and computer music.

3.3 The Musical Sessions for the Evaluation of the Applications

The 4 one-hour musical sessions took place in the ULIS classroom. A, B and C participated in all the sessions; D did not take part in the third one. The teacher in charge of the ULIS class was involved in all the sessions, whether the teaching assistant only in the third one. The professionals supported the pupils in the activities and helped them to maintain the concentration. The following applications were assigned to the pupils, on the base of the preferences expressed during the first interview:

- A: Didgeridoo (tested on smartphone), The Sea and Soundscape (web version tested on computer, and Max/MSP version tested on AKAI MIDImix controller).
- B: Soundscape and Djembé (tested on computer).
- C: The Sea (tested on smartphone), Soundscape (web version tested on computer, and Max/MSP version tested on AKAI MIDImix controller).
- D: Djembé (tested on computer), Didgeridoo and Wind (tested on smartphone).

During the first session, MO presented the aims of the study and provided pupils with basic knowledge on soundscape and electro-acoustic music. Afterward, an activity was carried out: firstly, the pupils had to listen to the explanations provided by the voice synthesis system, and to read the instructions in the downloadable file (see Figure 2). Then, they had to test the assigned applications on the basis of the instructions received. This activity continued throughout the second session. During the third and the fourth sessions the pupils created an imaginary story, through a brainstorming process based on the use of sticky notes. The story was then sonified with the applications: with the help of MO, the pupils wrote a musical score using a simplified scheme based on blocks of different colors. Each block represented a sound. The piece was played several times by the pupils using the applications.

3.4 Data Collection and Analysis

The pupils were involved in 2 face-to-face semi-structured interviews, before the beginning (questions 1-6, Table 1) and at the end of the session series (questions 7-11). For question 8 some visual aids [8] were used, to help pupils express their feelings. The professionals participated in a face-to-face semi-structured interview (questions 12-29), at the end of the session series. All interviews were audio-recorded. MO conducted the interviews and a participant observation of the musical sessions.

The evaluation framework used in this study assessed the following properties:

- pleasure generated by the use of the applications (assessed through observation and interviews with pupils);
- impact of the applications on the creation and performance of the musical composition (assessed through interviews with pupils);
- accessibility of the interfaces (assessed through observation and interviews with the professionals);
- adaptability of the applications to the context and to the users' needs (assessed through observation and interviews with professionals);
- playability (assessed through observation);
- sound quality (assessed through observation).

This framework was conceived on the basis of the literature [7, 9, 10] and the results of the evaluation conducted in 2021.

Data were analyzed using qualitative content analysis. The interviews were transcribed verbatim and the transcripts were returned to the professionals for feedback. The transcriptions and observation field notes were analyzed and sorted into categories and sub-categories.

3.5 Ethics Statement

The study protocol was approved by the ethics committee "Terre d'Éthique", University Hospital Center of Saint-Etienne, France (ref. IRBN882023/CHUSTE), on August the 21st, 2023. Informed consent was obtained from the pupils and their parents, as well as from the professionals.

4. RESULTS

4.1 Pleasure

All pupils expressed the wish to try musical applications (question 5, Table 1), because of the desire to discover something new and to learn music. All pupils liked making music with the applications (question 7), mainly because of the sounds they produced (2 answers). The most recurrent feelings generated by the use of these ADMIs were "excited, joyful, delighted, satisfied" (question 8). A asked several times to continue using the applications after the session series.

Initially, all the pupils wanted to use the smartphone (question 6); the reasons given by 3 of them were: "I have the habit of using a smartphone", "4G makes me want to use it" and "it's easier". 2 pupils mentioned the computer, and only 1 the MIDI controller, as second choice. 2 participants changed their opinion during the sessions, and expressed the wish to test the MIDI controller. At the end of the musical session series, 2 pupils affirmed (question 9) that they preferred to use digital applications for the music composition, rather than traditional musical instruments.

[8] The "emotions cards", see https://www.cres-paca.org/.

Pleasure (interviews with pupils)
1. Have you ever played music in your life?
2. What musical instruments did you play?
3. Have you ever used software or apps to play or create music? Which ones?
4. Did you enjoy using these software or apps?
5. If you never used them, would you like to try them?
6. If you could choose between playing music with a smartphone, a computer or a MIDI controller, which device/s would you choose?
7. Did you enjoy playing with these apps?
8. Can you describe the emotions you felt playing music with these applications?
9. What would you have preferred to use between apps and traditional musical instruments?
Impact of technologies (interviews with pupils)
10. Do you think that digital technologies made the composition easier to play, or would it have been easier to play traditional instruments?
11. How did technologies help you?
Accessibility (interviews with professionals)
12. Do the speed settings on the voice synthesis device fit users' needs?
13. What kind of changes would you make to improve the voice synthesis device?
14. Is the text on the interfaces legible for the pupils on computers and smartphones? Can you explain why?
15. What improvements would you make, to make the text easier to understand?
16. Is the text on the operation mode readable for the pupils on computer and smartphone?
17. What changes would you make, to make the text easier to read?
18. Are the explanations in the text understandable for the pupils? Can you explain why?
19. What changes would you make, to make the explanations easier to understand?
20. Do the labels and pictograms on interfaces help to understand the functions of the elements?
21. What changes are needed to make the functions of the elements easier to understand?
Adaptability (interviews with professionals)
22. Are the sizes of the interfaces and the elements suitable for the users' needs?
23. Which interaction mode (mouse, touchscreen, smartphone movement, Midi controller) best suits the the pupils' needs? Why?
24. What are the potential barriers and strengths of each interaction mode?
25. Which type of device (computer, smartphone, Midi controller) should be promoted for this kind of activities in schools?
26. Do you think web applications are appropriate for the school setting?
27. What are the advantages and disadvantages of the web applications used?
28. What other types of applications would you recommend?
29. Would you like to add any elements?

Table 1. Interview guides

4.2 Impact of the Applications on the Musical Composition

The 4 pupils affirmed that the applications made the work on the musical composition easier, compared to traditional musical instruments (question 10, Table 1). The reasons of these answers concerned the possibility of triggering sound by clicking on a button or moving the smartphone, and the fact of having the habit of using a smartphone (question 11).

4.3 Accessibility

- Voice synthesis system (questions 12-13, Table 1): for the professionals, the interface of the voice synthesis system was accessible and the voice was understandable; the playback settings allowed the speed to be adapted to a wide range of user profiles, but at minimum speed, the voice became unintelligible to pupils.

- Labels/icons on UI elements (questions 14-15): according to the professionals, the texts of the labels and the icons on the UI elements of the web applications were legible for the pupils, even on small screens.

- Instructions (see Figure 2) on the operation mode of the web applications (questions 16-19): for the teaching assistant "the text, with the adaptations for dyslexic pupils and the colors, is very accessible". The content of the text was "understandable", but pupils "may not be able to decode some terms" (such as “filter” or “effect”); this was confirmed by the observation. For the professionals, small images should have been added to explain what certain words referred to. In their opinion, the presence of an adult was necessary to help pupils understand and maintain the concentration.

- User interfaces (questions 20-21): on a general level, the professionals found the web applications interfaces accessible, and highlighted the importance of icons and labels to help understanding the UI functionalities.

4.4 Adaptability

4.4.1 Adaptability to the Users' Needs

- Interfaces adaptability (question 22, Table 1): for the professionals, the size of the interfaces and of their elements allowed a proper interaction with the web applications on all screen types.

- Interaction mode/gestures (question 23-24): for the professionals, touchscreen-based devices are the most suitable for pupils. Pupils use smartphones every day and can move their fingers very quickly on touchscreens. However, for the teaching assistant the choice of the interaction mode "depends on pupils' individual desires" and "it is beneficial to be able to propose the four options".

4.4.2 Adaptability to the Context

For the professionals, tablets and computers are the most suitable devices for school settings (question 25, Table 1). Tablets, in particular, are more suitable as they allow pupils to move according to the characteristics of the room. Moreover, for the professionals every pupil should have the same material, to avoid social inequalities. French schools

are normally equipped with tablets and computers, whether not all the pupils have the same smartphone or the same type of internet connection. This also emerged from the observation: only 1 pupil had an unlimited connection, 2 had a limited number of gigabytes and 1 had no connection. To use the applications, 2 participants had to ask the others to share the connection, causing distraction and quarrels. The professionals added that the Akai MIDImix controller could be another suitable device for pupils with NDDs: the movements of the faders and of the knobs are easy to control, because of their tangible dimension, and the object is attractive. However, schools could not provide MIDI controllers for every student.

According to the professionals, web applications are suitable for school settings (question 26), because no installation is required. However, some technical constraints emerged during the sessions: the access to some websites is not authorized by the French Education System. Therefore, an authorization was asked by the teacher to access the MO's website, where the applications were located. On the pupils' smartphones the access to some websites was forbidden by the parents. In addition, not all web application features are compatible with all browsers. Google Chrome is the only one that supports all the functionalities of the applications tested, but it was not installed on the school's computers: as they were Microsoft computers, only Microsoft Edge was installed by default.

4.5 Playability

Concerning Djembé, B showed difficulties in understanding how the sequencer worked. The concept of time sequencing was perhaps too abstract and complex for a pupil with intellectual disability. As regards The Sea, the movement to trigger the waves appeared not intuitive for A and C: they did not understand that the smartphone had to rotate around only 1 axis, and rotated it on multiple axes. In Didgeridoo and Wind, D succeeded in modulating the sound parameters as the multi-axis rotation allowed to do it. Soundscape appeared the easiest application to use, as sound triggering was based on buttons to press. No particular issues emerged on the use of the Akai MIDImix controller.

4.6 Sound Quality

As described in section 4.1, the sounds generated by the applications seemed attractive to pupils (question 7, Table 1). The only issue concerned Didgeridoo: the amount of distortion should be reduced, to make the sound more pleasant and realistic. The smartphone has a very low volume and needs to be amplified to be balanced with the other devices (amplified by speakers during the sessions). In Soundscape, not all the sound levels were balanced.

5. DISCUSSION

This article reports on the participatory and iterative design process and evaluation of some musical applications for pupils with NDDs. The involvement of the pupils and the professionals made it possible to collect relevant insights, that may benefit other researchers in this field.

The use of the applications seemed to elicit a feeling of pleasure in the 4 pupils: they liked playing with these ADMIs and expressed the wish to test them. The smartphone and the MIDI controller were their favourite devices.

As affirmed by the pupils, the applications had a beneficial impact on the music composition work, making it easier compared to traditional musical instruments. This is coherent with the results of the evaluation carried out in 2021 (section 2.2) [29]: indeed, all the participants involved in this study stated that the use of ADMIs greatly facilitated the work on the musical composition.

For the professionals, the adaptations described in section 2.2 made the interfaces more accessible for the pupils, but the presence of an adult is still necessary to let them use these ADMIs. This consideration highlights the role of teaching assistants [33] in musical activities with pupils with NDDs. This topic was addressed by another study [34], which underlined the crucial support provided by teaching assistants in SEN settings (particularly, in understanding students' needs and engaging children in the musical interaction).

Regarding adaptability, for the professionals the touchscreen-based devices, and in particular the tablets, are the most suitable for school settings. Pupils have the habit to use touchscreens and their use reduces inequalities: as French schools are equipped with tablets, every pupil would have the same material. This is consistent with another study [11], which confirmed the availability of tablets in SEN schools and the growing trend of using touchscreen devices. The professionals also emphasized the accessibility of the AKAI MIDImix controller, because of its tangible dimension. This is in line with what affirmed by Förster and Komesker [35], who highlighted the power of tangible interfaces to facilitate embodied interaction and reduce the cognitive load of users with SEN. However, for the professionals, schools could not provide MIDI controllers for every pupil, due to monetary constraints. The issue concerning schools' financial constraints for the purchase of ADMIs was also raised by other researchers [11, 34]. The professionals stressed the importance to provide pupils with several options, to fit different users' needs. This consideration seems in line with another study [11]: indeed, one of the teachers interviewed in this study underlined the need to have a multitude of ADMIs at school, so that students have the chance to find the instrument suiting their abilities best. The professionals added that web applications are suitable for school settings; however, some technical issues emerged from observation: the browser compatibility of the Web Audio API, stressed in the literature [36], and the fact that the French Education System does not allow access to the websites where applications are located.

As regards playability, one of the main difficulties observed concerned the use of the sequencer on Djembé. This seems not consistent with what affirmed by Förster and Komesker [35], who emphasized the accessibility of sequencer-based instruments: for them, the loop process

reduces the stress of children with SEN, who are not obliged to perform exactly in time. Another difficulty concerned the 1-axis rotation of the smartphone in The Sea. This also seems inconsistent with what affirmed by other authors [25, 37], who underlined the potential of accelerometer-based applications in SEN settings. To overcome these difficulties, more than 4 sessions should have been perhaps implemented for the evaluation.

The interviews with pupils also highlighted that the sound quality influenced the pupils' desire to use the applications. Also this issue was addressed by other authors: Ward and Davis [7] stressed the importance of ADMIs sound quality; moreover, in the above-mentioned study of Förster [11], the teachers interviewed highlighted that ADMIs should sound as pleasant as possible, so that it worst listening to for the students.

6. LIMITATIONS

A first limitation was due to the setting and the audience of the study: the ADMIs evaluated in this article are used in a school program aimed to pupils with NDDs. Because of the schools' constraints, and the variety of profiles that children with NDDs can have, it is not possible to develop customized instruments that fit individual needs [11, 34]. Another limitation concerned the fact that the pupils were not involved in the design process since the ideation phase [13]. As explained in section 2.2, the audience of the project and the context were identified more precisely at a later stage of the research. The participation of the pupils in the ideation phase might have provided further insights for the adaptation of the prototypes. Other limitations concerned the evaluation method: due to logistical constraints, the researcher who conducted the musical sessions and data collection was the same who carried out data analysis. This may have influenced the objectivity of the evaluation. In addition, the way interviews were structured did not seem appropriate for pupils with intellectual disability or communication impairments: indeed, B did not answer to several questions.

7. CONCLUSIONS AND FUTURE WORK

The considerations emerged from this study will guide the next iterations of the design process of the applications. The sound of The Sea will be improved, and the volume levels of the applications on all devices will be balanced. The playback settings of the voice synthesis will be adjusted, and more images will be added to the instructions (see Figure 2). Wherever possible, web applications should also be controllable via a MIDI controller. In the next evaluation the web applications will be tested on tablets and computers, and more than 4 sessions will be implemented; the participation of teaching assistants will be encouraged and to promote social inclusion and empowerment, groups of pupils with NDDs and neurotypical children will be involved. There should be a division of tasks between the persons carrying out data collection and analysis. More visual aids will be used in interviews with pupils.

Acknowledgments

The authors would like to thank all the pupils and the professionals of the Waldeck-Rousseau middle school, involved in the study. The authors would also like to thank the Maison des Sciences de l'Homme of Lyon-Saint-Etienne, the IXXI Rhône-Alpes Institute for Complex Systems and the Byosil - Systems Biology Network of the University of Lyon, for funding this study.

8. REFERENCES

[1] E. Frid, "Accessible Digital Musical Instruments-A Review of Musical Interfaces in Inclusive Music Practice," in *Multimodal Technol. Interact.*, 2019, p. 57.

[2] E. G. Duarte, I. Cossette, and M. Wanderley, "Analysis of Accessible Digital Musical Instruments through the lens of disability models: a case study with instruments targeting d/Deaf people," in *Front. Comput. Sci., sec. Human-Media Interaction*, 2023.

[3] M. Oliver, "The social model of disability: thirty years on," in *Disability and Society*, 2013, pp. 1024–1026.

[4] K. Samuels, "The Meanings in Making: Openness, Technology and Inclusive Music Practices for People with Disabilities," in *Leonardo Music J.*, 2015, pp. 25–29.

[5] J. Barbosa, J. Malloch, and M. Wanderley, "What does "Evaluation" mean for the NIME community," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Baton Rouge, LA, 2015, pp. 156–161.

[6] F. Lindetorp, M. Svahn, J. Hölling, K. Falkenberg, and E. Frid, "Collaborative music-making: special educational needs school assistants as facilitators in performances with accessible digital musical instruments," in *Front. Comput. Sci., Sec. Human-Media Interaction*, 2023.

[7] A. Ward and T. Davis, "Design Considerations for Instruments for Users with Complex Needs in SEN Settings," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Copenhagen, 2017.

[8] K. Graham-Knight and G. Tzanetakis, "Adaptive Music Technology: History and Future Perspectives," in *Proc. 41st Int. Computer Music Conference*, Denton, 2015, pp. 416–419.

[9] N. Davanzo and F. Avanzini, "A Dimension Space for the Evaluation of Accessible Digital Musical Instruments," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Birmingham, 2020, pp. 214–220.

[10] E. Frid, *Diverse Sounds: Enabling Inclusive Sonic Interaction*. PhD Thesis, 2019.

[11] A. Förster, "Accessible Digital Musical Instruments in Special Educational Needs Schools – Design Considerations Based on 16 Qualitative Interviews with Music Teachers," in *Int. J. of Human-Computer Interaction*, 2023, pp. 863–873.

[12] E. Zayas-Garin and A. McPherson, "Dialogic Design of Accessible Digital Musical Instruments: Investigating Performer Experience," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Auckland, 2022.

[13] C. Quintero, "A review: accessible technology through participatory design," in *Disability and Rehabilitation: Assistive Technology*, 2022, pp. 369–375.

[14] J. Charlton, *Nothing about us without us*. University of California Press, 1998.

[15] E. Frid, C. Panariello, and C. Núnez-Pacheco, "Customizing and Evaluating Accessible Multisensory Music Experiences with Pre-Verbal Children-A Case Study on the Perception of Musical Haptics Using Participatory Design with Proxies," in *Multimodal Technol. Interact.*, 2022, p. 55.

[16] J. Nielsen, "Iterative user-interface design," in *Computer*, 1993, pp. 32–41.

[17] P. Reimer and M. Wanderley, "Embracing Less Common Evaluation Strategies for Studying User Experience in NIME," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Shangai, 2021.

[18] A. Lucas, J. Harrison, F. Schroeder, and M. Ortiz, "Cross-Pollinating Ecological Perspectives in ADMI Design and Evaluation," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Shangai, 2021.

[19] M. Rodger, P. Stapleton, M. van Walstijn, M. Ortiz, and L. Pardue, "What Makes a Good Musical Instrument? A Matter of Processes, Ecologies and Specificities," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Birmingham, 2020.

[20] M. Wanderley and N. Orio, "Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI," in *Computer Music J.*, 2002, pp. 62–76.

[21] R. Jack, J. Harrison, and A. McPherson, "Digital Musical Instruments as Research Products," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Birmingham, 2020.

[22] D. Morris-Rosendhal and M. Crocq, "Neurodevelopmental disorders-the history and future of a diagnostic concept," in *Dialogues in Clinical Neuroscience*, 2020, pp. 65–72.

[23] L. Kossyvaki and S. Curran, "The role of technology-mediated music-making in enhancing engagement and social communication in children with autism and intellectual disabilities," in *J. of Intellectual Disabilities*, 2020, pp. 118–138.

[24] S. Campbell-Philips, "The Effects of Music On The Outcome of Learning On Children With Disability," in *The Int. J. of Humanities & Social Studies*, 2020, pp. 53–65.

[25] B. Farrimond, D. Gillard, D. Boutt, and D. Lonie, *Engagement with technology in special educational & disabled music settings*. Youth Music, 2011.

[26] A. Förster, C. Komesker, and N. Schnell, "SnoeSky and SonicDive – Design and Evaluation of Two Accessible Digital Musical Instruments for a SEN School," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Birmingham, 2020, pp. 83–88.

[27] WHO, *Skills for Health: skills-based health education including life skills: an important component of a child-friendly/health-promoting school.* WHO, 2003.

[28] S. Ren, S. Letz, Y. Orlarey, R. Michon, D. Fober, M. Buffa, and J. Lebrun, "Using Faust DSL to Develop Custom, Sample Accurate DSP Code and Audio Plugins for the Web browser," in *Audio Eng. Soc.*, 2020, pp. 703–716.

[29] M. Olivo, "Les instruments de musique numériques pour les personnes en situation de handicap cognitif et mental : exemples d'applications via le web," in *Proc. Int. Conf. Sound and Music Computing*, Saint-Etienne, 2022, pp. 643–652.

[30] WHO, *Promoting health through schools. Report of a WHO Expert Committee on Comprehensive School Health Education and Promotion. WHO Technical Report Series N°870*. WHO, 1997.

[31] J. Tejada, A. Murillo, and M. Berenguer, "Acouscapes: A software for ecoacoustic education and soundscape composition in primary and secondary education," in *Organised Sound*, 2023, pp. 1–9.

[32] C. Allaire, J. Ruel, and A. Moreau, *Communiquer pour tous. Guide pour une information accessible.* Santé Publique France, 2018.

[33] U. Sharma and S. Salend, "Teaching Assistants in Inclusive Classrooms: A Systematic Analysis of the International Research," in *Australian J. of Teacher Education*, 2016, pp. 118–134.

[34] M. Svahn, J. Hölling, F. Curtson, and N. Nokelainen, "The Rullen Band," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Shangai, 2021.

[35] A. Förster and M. Komesker, "Loopblocks: Design and Preliminary Evaluation of an Accessible Tangible Musical Step Sequencer," in *Proc. Int. Conf. New Interfaces for Musical Expression*, Shangai, 2021.

[36] B. Matuszewski and O. Rottier, "The Web Audio API as a Standardized Interface Beyond Web Browsers," in *J. Audio Eng. Soc.*, 2023, pp. 790–801.

[37] J. Quian and J. Sahagun, "An Application of Bluetooth-Enabled Accelerometer-Based Movement Detection for Musical Interactions with Mobile Devices," in *Proc. 3rd Int. Conf. Advances in Computing & Information Technologies*, Dubai, 2023, pp. 26–27.

Biosensors in Immersive Music Performance: *Andromeda* for Flute and Biofeedback

Penelope Bekiari
National and Kapodistrian University of Athens,
Department of Digital Arts and Cinema
p.bekiari@dcarts.uoa.gr

Anastasia Georgaki
National and Kapodistrian University of Athens, Department of Music Studies
georgaki@music.uoa.gr

ABSTRACT

In recent years, within the field of music technology, one of the fundamental topics that has engaged composers and researchers is the relationship between sound and control. Moreover, there is an increasing integration of biotechnology in interactive music performances. The piece *Andromeda* aims to utilise the biometric data of performers to enhance the sonic landscape, providing a multifaceted and deeply resonant dimension to the musical experience. This occurs notably during live performances and also within the context of theatrical roles, amplifying the artistic expression. This project not only emphasises the measurement of the human body during live music performances but also emerges within the framework of post-humanism. Its aim is to capture bodily data to track the performer's emotional changes during live performances. By mapping real-time physiological data onto sound processing parameters, the piece blurs the lines between performer and instrument, inviting a dynamic interplay between human expression and computational technology. Through a mixed media composition for flute and live electronics, *Andromeda* delves into the intricate relationship between theatrical conventions, emotional states, and musical narrative. By leveraging galvanic skin response, heart rate, and respiratory data, the composition transforms emotional fluctuations into immersive sonic experiences, fostering a deeper connection between performer and musical improvisation. The research methodology involves collecting biodata from individuals through biosensors while eliciting emotional responses, enabling the correlation of physiological indicators with emotional states. In performance, the piece's sound architecture responds dynamically to the performer's emotional fluctuations, creating a fluid sonic landscape trying to mirror the performer's inner state.

1. INTRODUCTION

Biotechnology in electroacoustic music primarily regards data that one can retrieve from a human body to map them on sound processing parameters for aesthetic purposes[1]. Blain, *in Biomusic: A novel technology for revealing the personhood of people with profound multiple disabilities*, states that biofeedback technology represents the future of control systems in various fields, including both art and science. This idea is not new; it has been evolving since the late 20th century and continues to evolve [2]. To further illustrate this, we can cite several examples of applications like the *BioMuse* system by Benjamin Knapp and Hugh Lusted in 1988, [3] that have captured the attention of the academic community due to their quality and aesthetics. For instance, the Conductor's Jacket designed by Teresa Marrin-Nakra and Rosalind Picard in MIT, 1998. Although the Conductor's Jacket was initially designed as a recording and monitoring device for scientific research, its ability to transmit data in real-time allowed Nakra to use it during live performances, where it functioned not as a passive monitoring device but as a body-instrument [4]. In the same logic, Michel Waisvisz's *The Hands* in the mid-1980s, progresses to Laetitia Sonami's *Lady's Glove* in the 1990s and 2000s, and also encompasses Atau Tanaka's BioMuse performances utilising electromyogram (EMG) signals of hand gestures. Today, artists are also experimenting with various tools that can easily provide them with data on computers, within environments like Max-MSP, where musicians feel they can engage in a dialogue with cognitive science, particularly data science, like Nino Liz Masclef. An example of this is seen in the creation of hypnagogic landscapes generated from EEG signals [5].

This paper presentation is about the work *Andromeda* (2023), a mixed media music composition for flute and live sound processing, inspired by the homonym tragedy of Euripides. The work is based on the use of data retrieved by bio-sensors placed on the body of the flautist. Moreover, the parameters of real-time sound processes are mapped linearly on bio-data deriving from the body of the flautist. This research revolves around examining the extent to which theatrical and performative conventions influence the emotional states of the performers; how the biometricall data retrieved from their reactions may contribute stimulatingly to the composition narrative, the development of the piece, and aesthetically. The core focus of the research is on understanding how incorporating real-time physiological data can enhance the immersive experience and emotional impact within the context of an interactive performance. This approach aims to bridge the realms of technology and musical expression, emphasising the interaction between performers and computational technology. Such exploration strives to contribute towards timbral exploration and artistic expression.

2. *ANDROMEDA* FOR FLUTE AND BIOSENSORS: AESTHETIC AND TECHNICAL ASPECTS

2.1. Aesthetics

While numerous studies explore the relationship between artistic practice and technology today, fewer delve into artistic expression within real artistic conditions[6]. This particular research project aligns with the fundamental concept of transhumanism. Technological posthumanism examines the profound implications of emerging technologies on humanity's future, navigating intersections of philosophy, science, and culture [7]. Grounded in the exploration of human enhancement, artificial intelligence, and ethical dilemmas surrounding technological progress, posthumanism envisions a future where humanity transcends current limitations [8]. Artists, in particular, are drawn to these themes, employing various mediums to engage with the complexities of posthumanist thought. This research aims to examine how the human body responds during the performance of an electroacoustic composition within a theatrical setting, specifically focusing on Euripides' *Andromeda*.

Exploring biometric data in music performance opens avenues for innovation and experimentation, enabling the creation of dynamic and responsive musical compositions. It is interesting to observe through electrode placement how the body responds to articulating music through bodily gestures[9]. Here, the performer's physiological state becomes an active component in shaping the evolving sonic landscape of Euripides' text, to explore how performers can control such data to enhance the musical experience by utilising expanded capabilities of the human body during music creation such as self-suggestion, concentration, and improvisation. Recent advancements in technology, particularly in artificial intelligence, machine learning and related fields, are ushering in an era of technological posthumanism characterised by human-machine symbiosis. The biosensors used for the making of this work were: a galvanic skin response sensor, a heart rate sensor and a 3D magnetic sensor for respiratory data. In this way, the particular work opens a dialogue with the idea of technological posthumanism, as it encourages the performing musicians to use contemporary technology to explore different ways of sound management with data derived from the human body.

Andromeda's text was chosen because it offered sonic landscapes that could create an auditory map but also a universal and timeless theme. This theme is none other than the theme of love, with which the majority of people usually identify. More specifically, the sonic themes in the soundscape, which is formed, are related to the story of Andromeda: echoes, the sensation of the waves surrounding her, and the wind. The music describes the first scene where Andromeda, tied to a rock and battered by the waves, speaks to her echo, pleading with the goddess of the night to allow her to live her love with Perseus. In this framework, Andromeda immerses the performer in audiovisual imagery, enabling them to extract from the narrative and generate corresponding biometric data reflecting varying dispositions. In conclusion, research endeavours (such as the one mentioned) hold the potential to influence various domains in music and theatre, extending to technology and the healthcare sector (music-sound/healing). To summarise, this research has transitioned from artificial intelligence to data technology (machine learning). Subsequently, sound processes are mapped to correspond to the performer's emotional fluctuations. Each emotional state is associated with a specific sound process, and since different performers yield different data, each performance becomes unique. The central goal is to explore the complexities of human emotion, viewing it as a unique phenomenon for each individual, and leveraging its potential creatively within computer-aided music composition and performance.

2.2. Technical Aspects and the Creative Process

The background research for the piece *Andromeda* was based on the collection of biodata from a group of 30 individuals. The data were retrieved from participants wearing biosensors while listening to music and observing images. Through a dialogue with an instigator, they were required to describe their emotions. This method facilitated the charting of a spectrum between different emotional states and their equivalent in data retrieved from the sensors: a galvanic skin response (GSR) sensor, a heartbeat sensor (HBS), and a respiratory sensor (RS). For example, an activation of the sweat glands results in moisture secretion through pores onto the skin, causing electrodermal activity. The alterations of positive and negative ions in the secreted fluid, is detected by the GSR as measurable changes in skin conductivity. Therefore the sensor's detection of changes in secretion results in data that can be mapped to underline different bodily states. Sweat secretion cannot be consciously controlled; rather, it is guided and balanced by our nervous system to respond to behavioural demands. The nervous system associated with emotional states represents a rapid response mobilisation system, facilitating immediate motor action. Increased emotional activity is linked to physiological indicators of "autonomic arousal," such as elevated heart rate, blood pressure, and sweating. For this reason, a respiratory and a heart rate sensor were also used in this study. While GSR is an ideal system for monitoring emotional arousal, it cannot directly reveal the emotion itself, i.e., whether one is emotionally stimulated by feeling pain, nostalgia, anger, love, or anxiety. This prompted us to monitor the function and rhythm of heart rate and correlate them with the values obtained from GSR. For instance, during a performance where the musician interprets a segment of the piece with a sense of nostalgia and concentration, it may register high values in the galvanic skin response sensor, surprisingly, indicating a state of calmness. Additionally, this calmness may coincide with a decrease in heart rate.

The sound architecture of the Andromeda was simple and the score easy to be read and understood by the musician. The main purpose was not to create a work with a deterministic character, but rather a piece based on the instructions and free interpretation. The tonality change controlled by the combination of measurements of GSR, heartbeat and respiratory sensors were applied more in the form of a choral effect, adding harmonics to the flute,

with resonant reflections. For example, the breath control sensor in combination with the heartbeat sensor in Max-MSP triggers many different reflections including a fourth, a seventh, an inverted fifth, a twelfth and interacts with the sequence change button of the reflections.The primary objective was to establish a dynamic interaction between Factor A and Factor B, fostering an internal cyclical flow as subsequent factors unfolded. Additionally, the GSR system played a pivotal role in influencing spatial values within the code, thereby shaping the spatial dimension of the piece. Moreover, it also impacted the sequence of chords, altering it based on fluctuations in the data captured by the heartbeat sensor. For instance, instead of following a conventional progression like A-C#-E-G#, the sequence could shift to E-C#-A-G# to reflect different levels of emotional arousal as depicted by the data.

The recorded values and observations from the sensors were set to MaxMSP as the minimum and maximum thresholds so that, during the piece, the performer could navigate between value boundaries that would create a sound processing, corresponding to their emotional state. The values of this measurement (1,100-3,500 MOhms-1 second per ohm to gauss = 898755224014.74 gauss- pair of electrodes placed close to each other) increased when there were changes in the individual's emotional state, particularly regarding anxiety levels. This is quite intriguing, especially when considering that during the musical performance of the piece in front of an audience, the baseline resistance value never dropped below 2 MOhms, while during periods of stress, it didn't remain below 3.500 MOhms. Specifically, some of the results collected during the flutist's calm state, with a one-minute difference, were: 2.655 to 3.120 MOhms, with corresponding heart rates of 85 to 125 bpm, respectively. Similarly, during intense stress induced either through performance anxiety and exposure to sound, or during preparatory exercises, the values fluctuated as follows: 3.500 to 4.100s, MOhms, with corresponding heart rates ranging from 98-145 bpm. During the flute performance, it became feasible, in combination with the other two sensors, to correlate the respiratory data with emotional stimulation. However, in general, it was observed that during high stress, tidal volume decreased due to rapid, shallow breathing, resulting in volumes less than 500 mL per breath. During moments of calmness, the typical respiratory rate ranged around 12-16 breaths per minute. The faster the musician breathed, the more it influenced her heart rate to some extent. Moreover, during pauses and moments involving extended techniques, such as playing with the keys of the flute, she made more body movements, consequently altering her breathing rhythm constantly.

2.3. Innovative Aspects in Biosensor Interactive Performance

The most intriguing aspect of this research lies in its exploration of biomusic and the role of the artist as a bridge between art and science. Here, the artist seeks to uncover methods for understanding and analysing the world while engaging in meaningful communication with others. Data analysis in music constitutes a narrative of reality. Artists and musicians may employ sonification techniques to craft musical compositions inspired by scientific data, environmental phenomena, or societal shifts, with the internal objective being both the exploration of these data and the deduction of scientific insights conveyed to the public through experimentation within the realms of art and music. Consequently, the methodology employed in this paper discerns, utilising biometric data, the alterations occurring in the body in response to real-life circumstances during the live presentation of an electroacoustic composition.

Figure 1. Intense stress rates in MOhms against heart rate in bpm**.**

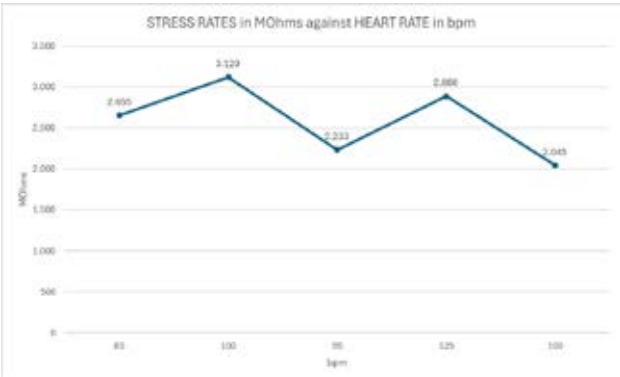


Figure 2. Stress rates in MOhms against heart rate in bpm.

The most intriguing aspect of this research lies in its exploration of biomusic and the role of the artist as a bridge between art and science. Here, the artist seeks to uncover methods for understanding and analysing the world while engaging in meaningful communication with others. Data analysis in music constitutes a narrative of reality . Artists and musicians may employ sonification techniques to craft musical compositions inspired by scientific data, environmental phenomena, or societal shifts, with the internal objective being both the exploration of these data and the deduction of scientific insights conveyed to the public through experimentation within the realms of art and music [11]. Consequently, the methodology employed in this paper discerns, utilising biometric data, the alterations occurring in the body in response to

real-life circumstances during the live presentation of an electroacoustic composition. The conditions of this experimental procedure include a live performance of a piece (individual's exposure in front of an audience improvising), the theatrical convention of Euripides' tragedy, and lastly, the significant influence emanating from the sound itself. To derive conclusive findings, the experimental process was iterated in reverse. By this, we refer to the inverse procedure where biometric data were sourced directly from the performer. Through a combination of focused concentration and self-regulation, the performer achieved a remarkable degree of control over the galvanic sensor and cardiac pulse readings. Specifically, within the context of significant engagements such as public presentations and theatrical performances (e.g., the tragedy 'Andromeda'), the performer demonstrated the ability to modulate sound processing parameters on the flute, including reverb, delay, chorus, and panning. The data obtained from this reverse procedure, particularly during the initial two minutes, exhibited consistent readings in the range of 4000-5000 MOhms, accompanied by corresponding heart rates exceeding 130 and 300-400 LSb (Likely Significant bits).

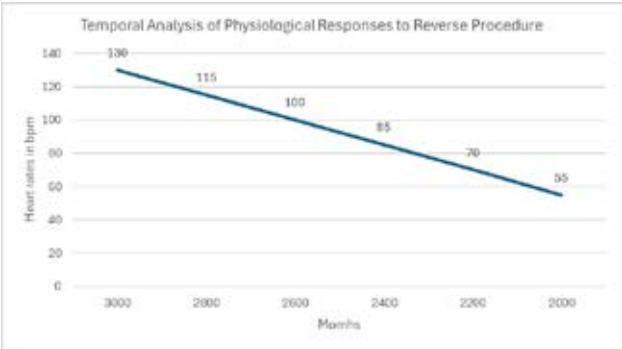


Figure 3. Temporal Analysis Physiological Responses to Reverse Procedure, heart rate in bpm against MOhms.

Subsequently, a steady decline was observed, with readings plateauing between 3000-2000, paralleled by reductions in both pulse rates and LSb values derived from the breath sensor. This reverse experimental approach serves to underscore a foundational tenet within contemporary discourse on technological posthumanism and cyborg theory. It illuminates how individuals, particularly since the latter half of the 20th century, leverage digital technologies to explore the capacities of the human body and express themselves artistically. Furthermore, from a methodological standpoint, this research endeavour not only contributes valuable data for interdisciplinary exploration, such as within the field of neuroscience but also enriches the realm of performative arts by fostering a symbiotic relationship between human biometric processes and artistic expression.[12][13]

3. CONCLUSIONS

Biomusic refers to the creation, manipulation, or interpretation of music based on biological processes, data, or principles, and relies on some kind of objectivity[14]. Embracing the fluidity between biology and technology, biomusic artists prompt listeners to reconsider their perceptions of sound, music, and the human condition, opening new avenues for creative expression and philosophical inquiry within posthumanist discourse. This research project concludes by discussing how the digital era's integration with nature has enhanced technological posthumanism, examining its impact on the arts and the broader community. In this particular research, the examination focused on whether practices such as technological transhumanism are capable of revealing the traditional musical experience [15]. Biometric data serves as a reflection of a reality occurring within the human body. However, how these data can be correlated with the performing arts is presented in two components.

The initial aspect entails real-time observation and data sonification. Observation links artists with science, emphasising scientific insights. Data sonification grants artists, generally, the liberty to interpret data critically. However, bio-music, exemplified by the Andromeda project, lacks this crucial freedom of movement, which we find significant.The objective was not the aesthetic of the sound form but the faithful representation of data at the moment of presentation. While performance thresholds were set, the primary aim remained to perceive the occurring changes, even if they occasionally deviated from anticipated aesthetic musical standards. As noted by Parker[16], data sonifications serve diverse purposes: representing real-time data, indicating alterations defined by designers within dataset-driven processes, and illustrating dataset trajectories. They notably influence user interactions with hands-on interfaces. For instance, Andrea Polli, a sound artist, employs sonification to translate data into sound, particularly in climate change installations, conveying complex atmospheric data to the public through immersive sound experiences. Referring to the examples, it is noted that the art of data sonification has a primary goal, which is to make scientific knowledge more accessible and understandable to the public. In other words, it aims to analyse and explain a truth while also enhancing it in an artistic manner[17]. Similarly, in the project presented earlier, *Andromeda*, the enhancement of the experience occurs both through the mapping of real-time data and through the theatrical analysis and representation of the work. Aesthetic considerations led to the selection of the sound design, which involves the flute with audio processing, to reflect the text of Euripides' lost tragedy. Specifically, as far as we know, in ancient tragedy, Andromeda symbolises both love and freedom, bound by her father and ready to be sacrificed to the sea monster sent by Poseidon. The waves strike her, and she pleads with the goddess of the night to allow Perseus to come and save her and love her. This was sonically rendered with intense reflections, exaggerated repetitions, recorded verses, and chorus effects. Andromeda in Euripides' work represents a symbol of defiance against patriarchy[18]and hope, an ancient Greek symbol that, rather than live a life she does not desire, prefers not to live at all. The choice to use the flute was made both as a means to connect with the ancient text, as ancient tragedies were commonly accompanied by the flute. Therefore, to conclude the components involving data sonification, the data transformed into sound includes both the performers' biometric data and the data extracted from the ancient text, in a mutually interactive format.

The second aspect delves into the convergence of technological augmentation and human expression in art,

particularly within the realm of performance arts. This intersection presents a rich ground for interdisciplinary exploration, with implications spanning neurobiology, psychophysiology, and performance studies. Contemporary discourse increasingly underscores the symbiotic relationship between technology and artistic expression, delineating terminologies such as "digital posthumanism" [19] to encapsulate this emergent paradigm. The integration of biometric data into theatrical and musical performances opens avenues for probing the neurobiological underpinnings of aesthetic experience and performer-audience dynamics[20]. The proliferation of sensor technologies capable of capturing physiological signals, including brainwave activity and autonomic responses, affords unprecedented insights into the physiological substrates of artistic expression. As the accessibility of biometric sensing devices expands, theatrical and musical practitioners are empowered to engage in empirical investigations of performance dynamics [21]. These endeavours encompass not only the elucidation of physiological correlates of artistic performance but also the exploration of therapeutic applications, such as stress modulation and psychophysiological self-regulation through artistic engagement [22]. For example, as the market opens up to sensors and brainwave measurement tools, including those used for this study, more theatrical and musical performances can utilise the function of biosensors. This results in small artistic experiments occurring under real conditions of pressure and creation, revealing not only how the human body reacts (biochemical changes) but also presenting a new method of studying music and, why not, a new way of stress therapy and body control through meditation. Besides the fact that the study of music can be enhanced through biosensor technology, electroacoustic music, due to the nature of its sound, can serve as a means of studying and understanding biometric changes. The present research aims to encourage efforts (using simple systems) like this, so that both data and observations can emerge during real performances, exploring more broadly the process of performance, both in theatre and music, and simultaneously how sound affects both the musician and the actor on stage.

Moreover, the domain of electroacoustic music emerges as a fertile ground for probing the perceptual and affective dimensions of biometric data. The inherent qualities of sound, as manipulated and synthesised in electroacoustic compositions, offer a unique lens through which to explore the complex interrelationships between auditory stimuli, physiological responses, and subjective experience. In sum, the research endeavours to catalyse interdisciplinary collaborations that leverage technological innovations to deepen our understanding of the physiological, psychological, and aesthetic dimensions of artistic performance [23]. By integrating empirical methodologies with artistic practice, scholars and practitioners seek to elucidate the mechanisms underlying human-machine interaction in the context of creative expression, thereby enriching both scientific inquiry and artistic discourse.

4. REFERENCES

1. Blum, T. (1989). [Review of Biofeedback and the Arts: Results of Early Experiments, by D. Rosenboom]. *Computer Music Journal*, 13(4), pp. 86–88. https://doi.org/10.2307/3679558.

2. Blain-Moraes, S. et al. (2013) “Biomusic: A novel technology for revealing the personhood of people with profound multiple disabilities,” AAC: Augmentative and Alternative Communication, 29(2), p.162 Available at: https://doi.org/10.3109/07434618.2012.760648.

3. Knapp, R.B. and Lusted, H.S. (1990) ‘A bioelectric controller for computer music applications’, *Computer Music Journal*, 14(1),pp. 50-60. doi:10.2307/3680115. Loeb Classical Library.

4. Bogaers, A., Yumak, Z., & Volk, A. (2020). Music-driven animation generation of expressive musical gestures. *Companion Publication of the 2020 International Conference on Multimodal Interaction*, p.22, doi:10.1145/3395035.3425244.

5. Schacher, J. (2022) 'Wearing a Second Skin of Sound: Touching the Other through Sounding and Listening', *Performance Research*, pp.36-37

6. Tervaniemi, M. (2007) “Review of The neurosciences and music II: From perception to performance,” *Music Perception*, 25, pp. 499-505

7. Cecchetto, D. (2013). N. Katherine Hayles and humanist technological posthumanism. In *Humanesis: Sound and Technological Posthumanism,* pp. 63–92. University of Minnesota Press. http://www.jstor.org/stable/10.5749/j.ctt46npm3.7.

8. Li, Junnan *et al.* (2024) ‘A domain generalization and residual network-based emotion recognition from physiological signals’, *Cyborg and Bionic Systems*, 5. doi:10.34133/cbsystems.0074.

9. Tanaka, A. (2011). Sensor-based musical instruments and interactive music. Oxford Handbooks Online [Preprint], p.352, doi:10.1093/oxfordhb/9780199792030.013.0012.

10. Jansen, T. and Väljamäe, A. (2021) ‘Exploring physiology-based interactions in performing art using artistic interventions / Kunstiliste Sekkumiste Kasutamine füsioloogiapõhiste interaktsioonide uurimiseks etenduskunstis’, *Methis. Studia humaniora Estonica*, 22(27/28), pp.1-20 doi:10.7592/methis.v22i27/28.18448.

11. Middleton, J. et al. (2023) “Data-to-music sonification and user engagement,” Frontiers in Big Data, 6., p.1, Available at: https://doi.org/10.3389/fdata.2023.1206081.

12. Khut, G. and Howard, C. (2020) ‘Mettāmatics’, Proceedings of the Fourteenth International Conference on Tangible, Embedded, and Embodied

Interaction, pp. 648–650. doi:10.1145/3374920.3375286.

13.Joseph, B.W. (2011) "Biomusic," Grey Room [Preprint], (45), p.1-130 Available at: https://doi.org/10.1162/GREY_a_00053.

14.Cheung, S. et al. (2016) "Biomusic: An auditory interface for detecting physiological indicators of anxiety in children," Frontiers in Neuroscience, 10(AUG), pp.2,8 Available at: https://doi.org/10.3389/fnins.2016.00401

15. Moore, L.J. and Moran, K. (2016) 'Posthumanism', *The Blackwell Encyclopedia of Sociology*, pp. 1–4. doi:10.1002/9781405165518.wbeosp063.pub2.

16.Parker, J. no. e (2018) "Composing [De]Composition: Data Sonification for Sound Art and Music Composition," Leonardo, 51(5), p. 13. Available at: https://doi.org/10.1162/leon_a_01664.

17.Mardakheh, M.K. and Wilson, S. (2022) "A Strata-Based Approach to Discussing Artistic Data Sonification," Leonardo, 55(5). Available at: https://doi.org/10.1162/leon_a_02257.

18.Blundell, S. (1995) *Women in ancient Greece*. London: British Museum Press.

19.Bogaers, A., Yumak, Z., & Volk, A. (2020). Music-driven animation generation of expressive musical gestures. Companion Publication of the 2020 International Conference on Multimodal Interaction, pp.22–26. doi:10.1145/3395035.3425244

20.Bazanova, O.M. et al. (2023) 'The biofeedback for musicians focused on ensuring students to award their peak performance', International Journal of Psychophysiology, 188, pp. 61–62. doi:10.1016/j.ijpsycho.2023.05.158.

21.Hynds, D. et al. (2024) 'Innermost echoes: Integrating real-time physiology into live music performances', Proceedings of the Eighteenth International Conference on Tangible, Embedded, and Embodied Interaction [Preprint]. doi:10.1145/3623509.3633356.

22.Gruzelier, J.H. (2018) 'Enhancing creativity with neurofeedback in the performing arts: Actors, musicians, dancers', Creativity Theory and Action in Education, pp. 223–245. doi:10.1007/978-3-319-78928-6_14.

23.Beira, J. and Quay, Y. (2014) BioMediation: A Biofeedback Audiovisual Performance in Proceedings: Presence, immersion and location: The 14th Biennial Symposium on Arts and Technology, February 27-March 1, 2014, Ammerman Center for Arts & Technology, Connecticut College, New London, CT. New London, CT: Ammerman Center for Arts and Technology, Connecticut College, pp. 1–5.

24.Reck Miranda, E. (2021) 'Handbook of Artificial Intelligence for Music', *Springer Nature Switzerland*, pp. x–xi. doi:10.1007/978-3-030-72116-9.

XTRACK: AUTOMATIC SEGMENTATION OF LIVE MUSIC WITH YAMNET

Florian COLOMBO (florian.colombo@gmail.com) [1,2], **Alain DUFAUX** [1], and **Davide PICCA** [2]

[1] **Lausanne Federal Institute of Technology (EPFL)**, Lausanne, Switzerland
[2] **University of Lausanne (UNIL)**, Lausanne, Switzerland

ABSTRACT

Recorded in UNESCO's Memory of the World Register, the collection of recordings of Montreux Jazz Festival concerts represents an invaluable audiovisual resource. Expert annotators have segmented the audio recordings to separate the music pieces from live concert events such as applause, speeches, and silences. This task is time-consuming and several rounds are necessary to homogenize the subjective decision of music borders. In this paper, we present XTRACK, an algorithm for automated segmentation of music pieces. It integrates the predictions of the sound classifier network YAMNET to create accurate segments. The algorithm is designed to isolate each track from live audio recordings. Finally, we evaluate the algorithm's performances on the human-labeled Montreux Jazz Festival archives.

1. INTRODUCTION

Recordings of live music are numerous and their number is increasing every day. To publish or archive such recordings, one typically needs to segment them into single music tracks: audio segments containing only music. Indeed, there are many moments without music from the moment the recording starts until it ends. Therefore, segmentation of live recordings aims to have a pleasant listening experience of individual tracks. To do so, student annotators were involved in indexing each live concert recording from the Montreux Jazz Festival (MJF). This constitutes a valuable database of audio segmentation.

In this paper, we present XTRACK, a deep-learning algorithm to assist people in segmenting live music recordings into individual music tracks. The segmentation algorithm employs the pre-trained audio classifier YAMNET [1]. Our contributions are the post-processing of YAMNET predictions to segment audio files in music and non-music segments and the performance evaluation on the MJF indexation data.

1.1 Montreux Jazz Festival audiovisual archives

Recorded in professional quality video and audio since the first edition in 1967, the audiovisual collection of the MJF gathers more than 5,000 concerts, representative of the greatest artists and musical trends of the last 50 years. It was inscribed in the UNESCO's Memory of the World Register in 2013, simultaneously with the creation of the Claude Nobs Foundation which is in charge of its preservation. From 2007, and in the frame of the so-called Montreux Jazz Digital Project [1], EPFL has been responsible for inventory, digitization, cataloging, storage, and enrichment of the collection. The Cultural Heritage and Innovation center of EPFL (CHC) was created for that goal, gathering all data and metadata in a dedicated relational database and defining numerous research, education, and innovation projects to enrich the collection.

1.2 Dataset: Cataloging and Indexing

The dataset represents more than 11'000 hours of video and 6000 hours of audio, digitized in uncompressed formats or directly captured to professional broadcast files in recent years. The sample rate of the digital audio stream is 96 kHz (24 bits precision) when digitized from analog support, but it can be 96, 48, or 44.1 kHz (16 or 24 bits) when captured in a native way from digital tapes. The associated Metadata consists of both technical information obtained from digitization (including hash and media information exports) and concerts' information (concert name, title of the pieces, hall, date, artists' names, associated instruments, authors' rights, original tape format)

Since 2013, a large project has been developed to catalog and index every concert, identifying series of events (introduction, song, speech, interlude, applause, encore, or silence), indexing precisely their respective start/end time locations in the media files, and verifying concert metadata (titles, artists, hall, year, etc). The motivation for this work is to be able to navigate comfortably across the 50'000 songs of the live concerts' collection, being able to properly and precisely playback them like in CDs or DVDs. Consequently, high precision is required in the determination of the start and end time indexes of every event. Indexing is performed by specially trained people, who need extensive experience of the multiple scenarios that can occur within song transitions. Such scenarios include

- The end of the song is sudden, a 1-2 seconds silence time occurs before the public starts applauding
- The end of the song can be easily identified and predicted by the public, resulting in a mix of musical song fade-out and applause fade-in period, followed 5-10 seconds later by a few words from a musician, speaking for a while about his last song

[1] https://go.epfl.ch/mjdp

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

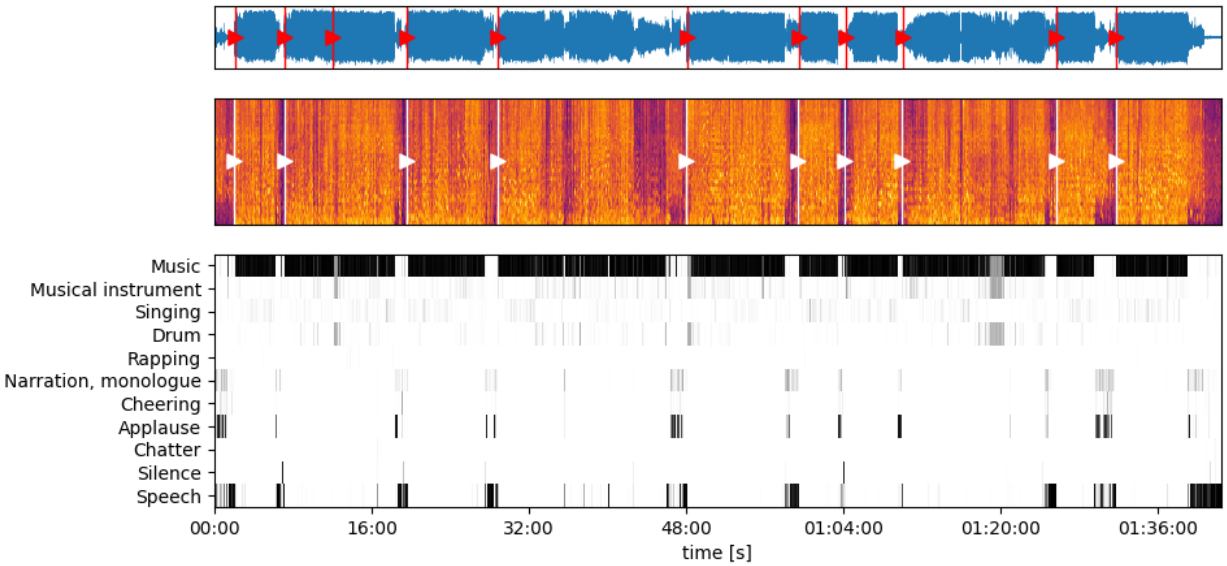


Figure 1. On top, a plot of the waveform from a live concert recording with track start labels from the MJF archives shown as vertical bars. In the middle, the frequency decomposition of the waveform with the XTRACK prediction for the starts of music segments shown as vertical bars. On the bottom, the normalized YAMNET predictions for the 5 music classes and 7 non-music classes.

- Same situation but with the musician speaking immediately at the very beginning of the applause, saying *Thank You!* and possibly presenting the musicians while some music goes on. Then the next song is chained
- The next song starts immediately during the fade-in of the public applause, which then rapidly stops. No speech from the band.
- No transition between two songs, a new one starts suddenly in the previous one, and the audience applauds for a short time while the new song starts.
- The song gets quiet and applause starts but a solo moves on. Applause fades out and stops rapidly, to start again over the music at the end of the solo. The music goes on for a while before its real completion.

Where to place the end and start time codes of every song? Criteria were defined for usual situations but the question is difficult and decisions can often be subjective. The indexing work requires extensive training before it can be performed reliably. Consequently, a detailed workflow was introduced with two phases, involving different documentalists, followed by mixed manual and automated quality checks, and weekly team discussions for solving difficult or uncertain situations.

2. RELATED WORKS

Trained on Google's AudioSet data comprising over 2 million annotated audio clip [2], YAMNET is a deep convolutional neural network based on the MobileNets architecture [3]. In particular, YAMNET processes audio files (16kHz sampling rate) and outputs at every 48ms frame a probability distribution over 521 classes (e.g.`speech`, `happy music`, `toothbrush`, ...).

In 2013, [4] proposed to automate part of the MJF indexation work by detecting occurrences of applause. Of course, applause does not correspond to song transitions in all cases but the possibility of a semi-automated approach was considered for the indexing work. New AI solutions such as YAMNET are promising, and the question is raised again. This time to fully try and mimic human work, taking thousands of hours of indexed concerts as a benchmark dataset, a new study was launched, and the first outcomes are described in this paper.

3. METHODS AND DATA

Whereas the MJF archives' indexation follows a well-defined protocol, many boundary decisions are subjective and do not directly correspond to audio events. For example, the end of music segments does not happen at the end of music but rather some flexible times later (from 0 to 8 seconds depending on the context and the subjective decisions of the human annotators). From a deep learning perspective, these observations directed us not to train a model directly on these subjective (inaccurate) data. Rather, we inquired how the audio classifier YAMNET [1] can be used for segmenting audio files and take advantage of the MJF data to validate the boundary decisions of XTRACK.

3.1 From YAMNET predictions to audio segments

We start by processing the audio of live concert recordings with YAMNET. To adapt YAMNET to our task, we constrain the probability distributions over the complete set of 521 classes to a smaller set of meaningful classes. XTRACK focuses on the 5 music classes and 6 non-music classes listed in Table 1. The temporal evolution of the

Description	Equation	Value
YAMNET music classes	C_{music}	[`Music`, `Musical instrument`, `Singing`, `Drum`, `Rapping`]
YAMNET non-music classes	$C_{non-music}$	[`Silence`, `Speech`, `Narration`, `monologue`, `Chatter`, `Cheering`, `Applause`]
Probability threshold to classify the nth frame of audio file x as music	`if` $\sum_{c \in C_{music}} Pr(x[n] = c) > \Theta_{music}$ `then` $n \rightarrow$ `music`	$\Theta_{music} = 0.2$
Probability threshold to classify the nth frame as non-music	`if` $\sum_{c \in C_{music}} Pr(x[n] = c) < \Theta_{non-music}$ `then` $n \rightarrow$ `non-music`	$\Theta_{non-music} = 0.1$
Consecutive music frames to place a music start index	`if` $m \in [n - L_{music} : n] \rightarrow$ `music` `then` $n \rightarrow$ `start`	$L_{music} = 20[s]$
Consecutive non-music frames to place a stop index	`if` $m \in [n - L_{non-music} : n] \rightarrow$ `non-music` `then` $n \rightarrow$ `stop`	$L_{non-music} = 4[s]$
Music start offset	`start`$[i] \rightarrow$ `start`$[i] + d_{start}$	$d_{start} = -0.1[s]$
Music stop offset	`stop`$[i] \rightarrow$ `stop`$[i] + d_{stop}$	$d_{stop} = 4[s]$

Table 1. Hyper-parameters of the XTRACK audio segmentation algorithm.

probability distributions over these classes (e.g. bottom plot of Figure 1) is the input signal from which XTRACK infers music segments.

Table 1 presents the fundamental operations of the XTRACK algorithm. To detect when the music starts, we look for frames `start`$[i]$ where the probabilities associated with the `music` class are above Θ_{music}=0.2 for L_{music}=20 consecutive seconds. After a music start frame has been identified, we look for the next time frame `stop`$[i]$ where the probabilities associated with the `music` class are below $\Theta_{non-music}$=0.1 for L_{music}=4 consecutive seconds. We iterate this process until the end of the audio file.

We then combine these boundary frames to segment the original audio file into music and non-music segments. According to the value of the class-constrained YAMNET predictions within the non-music segments, we predict its content (speech, applause, or silence) to label the predicted segments automatically.

For a more pleasant playback of segmented music tracks, we index the start of a music segment $d_{start} = -0.1[s]$ before the predicted `start`$[i]$ and $d_{stop} = 5[s]$ after the detected `stop`$[i]$.

3.2 Evaluation

To evaluate the segmentation decisions of XTRACK, we compare the algorithm results with the indexed database of the MJF archives. In particular, we look at how the onsets of predicted and human-labeled segments differ. In the Results section, we report the statistics of these timing differences.

Because of the subjective nature of the indexation process, we also conducted a qualitative evaluation of the segmentation algorithm. To do so, we randomly select examples where the XTRACK predictions and the MJF data differ and report the audio contexts before and after these segment onsets.

4. RESULTS AND DISCUSSION

To generate the following results, we applied our segmentation algorithm to 10 years (from 2008 to 2017) of MJF live recordings. It takes less than 1 minute for XTRACK to process 1 hour of audio on a MacBook Pro Retina 2013 (CPU 2.6GHz Intel Core i5, 16GB RAM).

4.1 Quantitative evaluation

In Figure 2 we reported the statistics of the timing differences between the automatic segmentation of XTRACK and the human-labeled MJF data. These values were found by aligning predicted and human-labeled indexes. XTRACK recovered 75% of the indexes. Out of every predicted start/stop, 75%/60% fall before the human-labeled timing and 25%/40% after. When XTRACK finds a music segment, 50% of the time it places its start/end within 0.9/2.9 seconds from the human-labeled index. These larger differences for the music stops are expected as the MJF stop indexes are placed with some flexibility (i.e. a few seconds after the end of the music). In summary, these statistical values suggest that the playback of individual music tracks, as determined by XTRACK, will be pleasant and respect the actual music boundaries of the live concert.

XTRACK decisions are based on several hyper-parameters (see Table 1). These values were found with trial and error to optimize the algorithm's quantitative and qualitative performances. In case the concert setlist is available (the number of individual tracks is known), an

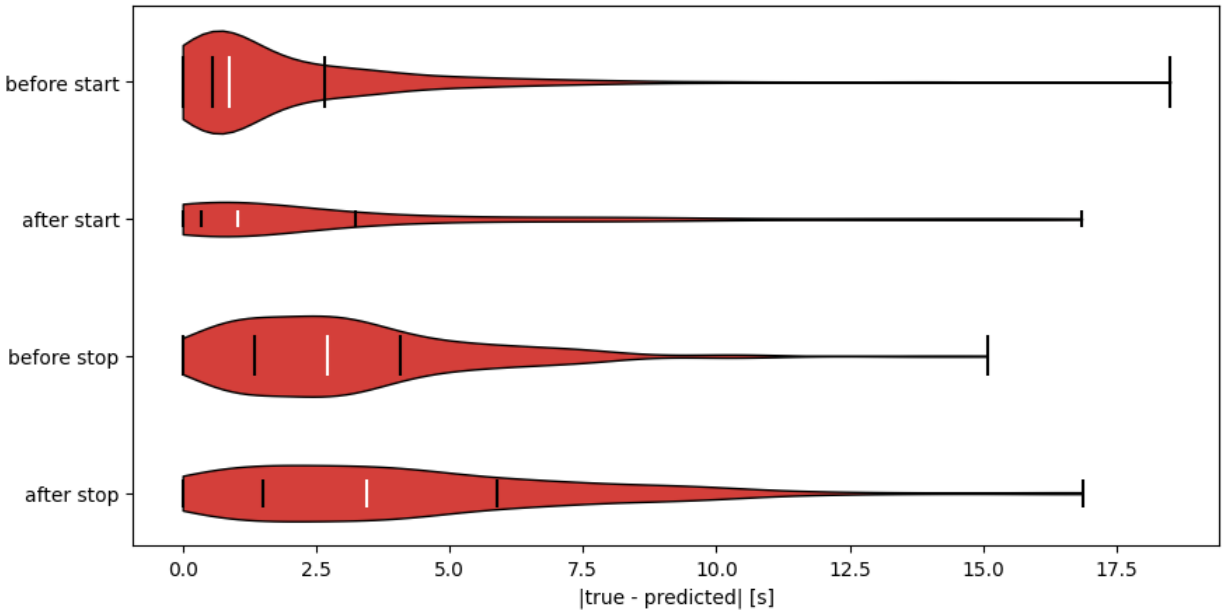


Figure 2. Densities of the timing differences between the predicted and human-labeled starts and stops of music segments. The density is computed using Gaussian kernel estimators of the python `matplotlib.pyplot.violinplot` function. To better understand the values, we chose to visualize separately the timing differences when the predicted transition is before (too early) or after (too late) the human-labeled index. The width of the densities represents the number of predictions falling in each category (before or after). The vertical bars indicate the two extrema and the [0.25 0.5 0.75] quartiles. The values in seconds are: before start [0.56, 0.87, 2.65], after start [0.36, 1.04, 3.25], before stop [1.35, 2.71, 4.07], and after stop [1.51, 3.45, 5.89].

automated grid search or a manual correction for the hyper-parameters values can be performed to find the correct number of tracks. For example, the number of consecutive non-music frames required to add a music stop index can be adapted to find fewer or more tracks.

4.2 Qualitative evaluation

In Table 2 we describe six randomly selected segments where there is a difference between the MJF indexes and the automated segmentation of XTRACK.

10% of predicted music starts cannot be aligned with the MJF data. These are boundaries found by XTRACK that are not present in the MJF data. Row 4 of Table 2 describes such an event. In that case, this additional index does not harm the pleasantness of the playback of individual music tracks.

More than 75% of `music` segments are found by XTRACK. The remaining includes track changes without interruption in the music. In particular, 25% of the `music` indexes from MJF data are music-to-music transitions. To cope with that, XTRACK sometimes adds a `stop` index between two consecutive `music` indexes of the MJF data. Indeed, 20% of the predicted `stop` could not be aligned with the MJF data and are therefore supplementary indexes where XTRACK predicted an end of a music track before the start of the next track. The MJF annotators sometimes decided to add a new `start` index without a `stop` before.

`start`$[i]$ at 10:00 in Figure 1 and rows 2+6 of Table 2 illustrate a continuous music-to-music transitions that is not detected. As XTRACK makes its boundary decisions based on when the music starts and stops, these transitions without interruption in the music cannot be detected. In these cases, the playback from XTRACK includes more than one track. However, as the music does not stop, it is still a pleasant cut to listen to. Additional analysis (e.g. by adding more classes to C_{music} and $C_{non-music}$) could help detect changes in musical contents and find these music-to-music transitions.

To sum up, many differences between the XTRACK segmentation and the MJF data can be explained. They can be due to subjective choices in the manual indexation of the Montreux Jazz Archive. As a final check, we randomly listened to individual tracks determined with XTRACK. These listening sessions were fluid and a more pleasant experience than listening to the full concert.

5. CONCLUSIONS AND FUTURE WORK

In this study, we introduced XTRACK, an innovative algorithm designed to automate the segmentation of live music recordings. By leveraging the sound classification capabilities of YAMNET, our approach has significantly streamlined the process of isolating musical segments from live concert recordings, addressing a critical need within the Montreux Jazz Festival archives and potentially other music collections.

For automated indexation of digital music (e.g. on YouTube) or as a tool to index musics in large collections, XTRACK can suggest a first segmentation. This proposition can be fine-tuned by a human checker or used as it is.

MJF	XTRACK	Description	Comment
08:12 `applause` 09:04 `music`	08:17 `speech` 09:03 `music`	music > 08:12 applause 08:30 speech 09:04 music	On one side, the MJF index starts at the very beginning of the 08:12 applause. On the other side, XTRACK adds 5 seconds of applause after the end of the music. There is a difference in the labeling as well: XTRACK predicts a `speech` segment. The MJF label is `applause`. There is indeed a 34s-long speech validating this XTRACK labeling decision.
24:39 `music`	`None`	music > 24:30 cheering 24:40 singing	A single high guitar note is held from 24:30 to 24:55. The music did not stop but there is a transition to a new track. Not stopping here do not disturb a pleasant playback but a new track begins.
01:05:30 `music`	01:05:41 `music`	silence > 01:05:30 music 01:05:36 silence 01:05:41 music	The silence between the small introduction and the rest of the track is too long for XTRACK to include it in the music segment. A playback starting as indexed by XTRACK (01:05:41) is still very pleasant.
`None`	01:27:28 `music`	cheering > 01:27:28 music	The music starting at 01:27:28 is a background music played from the concert hall loudspeakers on top of some crowd noises. Whereas, this is not a concert track, our algorithm predicted a `music` start index. However, that is because YAMNET detected the background music. Adapting the Θ_{music} can be made to be more strict and might be a way to prevent such mistakes. However, listening to a few seconds of this segment is sufficient to throw it away.
40:10 `applause` 43:10 `music`	39:56 `speech` 43:10 `music`	music > 40:00 cheering 40:45 quiet 42:30 speech 43:10 music	This is a pre-recorded electronic music concert. The music is displayed through loudspeakers in the hall. At 39:50, the electronic music outputs recorded applause that confuses our algorithm, which cut earlier than the MJF data. This cut is valid for a pleasant playback with a smooth fade-out. Moreover, the `speech` prediction fits the audio context.
06:14 `music`	`None`	music > 06:14 music	Transition to a new track without interruption not found by XTRACK.

Table 2. Qualitative description of six randomly chosen discrepancies between XTRACK decisions and the human-labeled MJF archives. The **Description** column illustrates the analysis of a musician listening to the original audio files. In the **Comment** column, we explain in words the context.

Its code is available on GitHub [5].

Specifically, the algorithm demonstrated a high success rate in accurately identifying and segmenting music tracks, thereby facilitating a more efficient archival process. The outstanding performance of XTRACK in music segmentation has inspired us to explore additional applications of this technology.

Given the rich emotional labels attached to YAMNET dataset, we propose extending the use of XTRACK beyond traditional music segmentation. By integrating these emotional labels, we aim to construct segments that not only identify different musical pieces but also classify the underlying emotions conveyed in each segment. This development opens up new avenues for emotional analysis within music, allowing for a deeper understanding of the impact of live music on audiences. In fact, this approach promises to enhance the user experience by providing more nuanced and emotionally resonant music listening experiences. Moreover, it can serve as a valuable tool for musicologists, psychologists, and other researchers interested in the interplay between music and emotions.

The XTRACK segmentation algorithm can be used for other interesting audio segmentation tasks. For example, we can detect in- and out-of-tune segments by analyzing the magnitude of high-frequencies of the spectrogram (fast Fourier transform `fft` of the audio files). By imposing a threshold on the amplitude of a high-frequency range, we can find the longest segment of in- or out-of-tune music. Used as a pedagogical tool, XTRACK segmentation algorithm is promising.

Acknowledgments

We thank the Cultural Heritage and Innovation Center (CHC) of École Polytechnique Fédérale de Lausanne (EPFL) and the Information and Language Section (SLI) of the Lausanne university (UNIL) for hosting this project. This project was possible thanks to the BNF national qualification program of Bern university (UNIBE). Finally, we are grateful for the recording work of the Montreux Jazz Festival, for its preservation by the Claude Nobs Foundation, and to all the musicians who contributed to this amaz-

ing concert collection.

6. REFERENCES

[1] M. Plakal and D. Ellis, "Yamnet," https://github.com/tensorflow/models/blob/master/research/audioset/yamnet, 2020.

[2] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 776–780.

[3] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[4] D. El Badawy, P. Marmaroli, and H. Lissek, "Audio novelty-based segmentation of music concerts," in *Acoustics 2013*, 2013.

[5] Florian Colombo, "Xtrack," https://github.com/FlorianColombo/xtrack, 2024.

Cyborgutt: Exploring the body's fluid boundaries through a biosensor composition

First Author

Fotis Rovolis (φø)
faux.tease@gmail.com
NKUA

Second Author

Areti Andreopoulou
aandreo@music.uoa.gr
NKUA

Third Author

Thanos Polymeneas-Liontiris
athpoli@gmail.com
NKUA

ABSTRACT

For Massumi the body has immaterial boundaries, being connected to space, time and movement. This paper presents an exploration of this notion through a creative-critical interactive music composition. The trajectories unfolded in this research meet in the cross over between queer politics and cyborg theory. The results of an initial theoretical research in this territory led into an interactive composition, a music work based on the practice of biomusic, of which the subject matter is the human body integrated into the current digital technological enhancements. The work explores the body and its sounds, through a genderless and fluid lens, to construct the soundscape of the body. Utilising biosensors and interactivity, the research highlights the way the body can be turned into an instrument for the making of its own soundscape. Acquiring breathing, heartbeat, and muscle sensors to modulate aspects of a body's soundscape, the experience of the performer is enhanced, by incorporating their somatic functions as modulators in the performance. Therefore, the research approaches the relationship between body and music through a creative-critical practice. It challenges certain stereotypical views on human nature, exploring a rather post-human narrative on the body. It observes it as an organism not strictly circumscribed inside flesh, but an organism of fluid boundaries and relations with its surroundings. The work aims to depict how interactive technologies used as tools in a cyborg art practice can contribute towards the desire to attain a cyborg identity. The paper describes the processes used in this creative-critical practice, its aesthetical aims and the bodily and technological challenges. Concluding, the research opens up a discussion on the experiential outcomes of the performance and proposes enhancements for a deeper exploration.

1. GENDER, VOICE AND BODY FLUIDITY

Judith Butler explains that gender is a social construct created by an exclusionary process that ends up trapping and oppressing bodies, and ultimately leading to their disappearance. According to Butler, gender is produced through a repetitive pattern of performed gestures and actions, a result of processes rather than something that predates, assigned at birth [1]. Within these processes, queer bodies that do not self-identify within the stereotypical gender binary spectrum (male - female) end up feeling invisible. This collective invisibility of existences leads to a series of temporally displaced experiences, creating alternative queer temporalities [2], a concept that responds to heteronormative standards and linear life milestones (such as adolescence, work, family), just as the broader concept of queer does: proposing constant fluidity, redefinition of boundaries, and recognising that queer bodies explore new relationships with time.

Stephen Conor and Joke Dame explore the role of the voice in gender attribution and performativity. Stephen Conor explains how every voice always implies a body, and correspondingly, every listening to the voice produces a "vocalic body" [3]. That is, as soon as one hears a voice, they immediately attribute it to a body, assigning the corresponding gender marker (figure 1). Thus, as Joke Dame suggests, the voice does not inherently possess any gender trace, but instead, it is the act of listening that attributes gender to each voice: perception operates within predetermined conventions. Therefore, she proposes that hearing should be reinvented every time one encounters a voice [3]. Additionally, for Jarman Ivens, the space in which gender boundaries can be deconstructed in the field of voice is the space between transmitter and receiver, what she calls *thirdspace*. It is the moment after the sound waves are transmitted from the body and before they are received by the hearing of another body, where the voice still contains infinite expressions, without yet acquiring a male or female signifier [4].

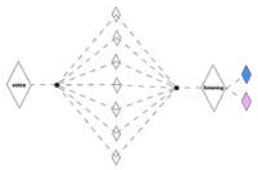


Figure 1. Attributing voices to bodies

Copyright: © 2024 First author et al. This is an open-access article dis- tributed under the terms of the Creative Commons Attribution License 3.0 Unported, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Further on regarding body's boundaries and fluidity, the work of Brian Massumi challenges traditional perceptions that see the body as a stable, passive entity. Instead he advocates for a relational understanding of the body. From this perspective, the body is considered as a dynamic, constantly evolving organism shaped by its interactions with the world, a world that is actively participating in shaping [5]. Massumi distinguishes the body's fluidity into three aspects or systems, where the predetermined conventions under which it operates, as in previous fields, function in non-inclusive manners, thus trapping the body in finite forms [6]. Massumi's first aspect of body fluidity is summarised through the term "incipience," the intermediate moment after a state, movement, or action is completed and before its next occurrence [7], suggesting that the body constitutes a continuum set of shape, leading to motion and performance. Stern describes the body as a condition in a permanent potential position for activating new forms [6], similarly to gender and voice. A set of correlations, rather than an isolated shape, which is also part of a system of relationships among the bodies. Donna Harraway describes the relationality between bodies, through acquired tentacles [8] as multi sensory organs, contributing to a unified network where senses and boundaries are shared, translating into mutual care and responsibility among all. Finally, when this system of relations is understood as an integral element of space, one witnesses a structure being constantly created. According to Stern, the only way to learn about the body is by using it [6]. The experience and sensation of the body for its relative determination in space is proprioception and it is an essential component for the development of this relationality. Through proprioception, one perceives the body's location, balance, speed, and the position of its parts.

These qualities that contribute to a fluid understanding of the body and its relationships create the background for Stern's efforts to redefine the concept of "interactivity" as a condition strengthened through deeper qualities of sensory perception [6]. This means that each individual literally "interacts" through the process of movement-emotion-thought as a material bodily process, rather than being a technological informational entity that determines its actions. A digital work can only be "foolishly" interactive, offering limited computational capabilities, but this seems to diminish as technology gains more meaning than the actual experience itself. As Butler, Halberstam, and Massumi explain regarding the complexity of gender, time, and body, when placed in the context fragmentation of information, quant[1]ification, and segmentation of characteristics, they end up losing their overall substance and they only function as closed fields of deterministic connections and behaviours. On the contrary though, any relationship with digital technology should be deconstructed through the spatiotemporal kinetic duration of a body that literally "interacts" with it.

In line with the above considerations regarding the boundaries of the body, resides the reconsideration regarding the human condition in Donna Haraway's Cyborg Manifesto [8]. In this emblematic work, the notion of cyborg comes to challenge previous definitions of the human subject. Haraway explains how the fluidity of gender and nature manages to redefine social structures, deconstructing the rules and beliefs upon which human essence is built. Ultimately, the existence of the cyborg manages to dismantle all opposing systems, human-animal, organism-machine, man-woman, through the loss of pure human dimension, which is now problematised [8]. According to Haraway, 'Cyborgs are about remaking and reserving against reproduction and birth. [...] We have all been wounded, deeply. We do not need re-creation and rebirth, but rather remaking, and the possibilities of our reconstitution are contained in the utopian vision of a genderless monstrous world' [8, pp 67]. Furthermore, Gillian Haddow introduces the term "everyday cyborg" [9]explaining that the cyborg is not something distant found only in fiction but something we already are. The use of medical implants (pacemakers, prosthetic limbs) for curing medical conditions and enhancing humans is a process that creates informal cyborgs. Jaime del Val defines technology as a continuous self-organisation of motion, thus highlighting how biology and technology intersect at the level of continuous modification and production of infinite variations, whether in bodies or in the systems they shape [10]. The Cyborg Foundation and the Transpecies Society, two groups of contemporary cyborg artists, are currently expanding the idea of everyday cyborgs; they believe that cyborg identity is not about the stage at which each person's body is, whether they have implants or not, but primarily about how the individual feels. "If you feel like a cyborg, then you are a cyborg. It's more about the sense of identity than anything else" [11]. Artist Neil Harbisson, one of the first cyborgs and a founding member of the Cyborg Foundation, born with achromatopsia, manages to convert the colours he sees into frequencies through an implanted sensor in his head. As they describe, before undergoing the implantation, when he still used the device as an external attachment to their head, their brain had already "learned" to hear colours, so that from a certain point onwards, seeing colours produced frequencies. This new sense (exosense) had become connected to Harbisson's body, even without the completion of the sensor implantation procedure in their head. Another example is a recent member of the cyborg community, musician Kai Landre. Landre has assembled and attached to their hand a sensor system that detects cosmic radiation. The frequencies of that radiation are mapped to notes which are emitted as an auditory signal through a transmitter placed on their head. With this mechanism, Landre receives continuous auditory information that corresponds to the analysis and sonification of cosmic rays, and translates it into musical compositions. The exosense system may be the ultimate tool for redefining the relationship between the musical instrument and the musician. Interaction takes place on a

[1] The term cyborg emerged from Manfred Clynes and Nathan Kline in 1960, blending the words cyber and organism to refer to an enhanced form of human with simultaneous human and technological characteristics. The concept of enhanced existence, resulting from the augmentation of existing psychological, cognitive, and physical abilities or the addition of new ones, forms the basis for the development of exosenses.

multiplicity of different levels and interpretations, it becomes cognitive, and ultimately it concerns the dialogue between the subconscious mental, cognitive, and behavioural states of the musician with the system. It ends up being the instrument itself [12].

2. BIOPHYSICAL MUSIC

Biophysical music is a type of music produced using biological data and physiological signals, such as heart rate, respiratory rhythm, and muscle activity. Its goal is to create a direct connection between the performer's physical state and the music production. In this way, biophysical music aims to blur the boundaries between performer and instrument, making the performer's body the instrument itself. Marco Donnarumma approaches biophysical music by using his own body as a source of musical data, measuring and processing physiological signals such as heart rate, muscle activity, and respiration. He then maps these signals to musical parameters such as pitch, intensity, and rhythm to control the sound production in real-time. This way, he creates compositions that reflect his physical state at any given moment. By using technology to process and enhance his physiological signals, Donnarumma is able to create musical experiences that would not be possible without such type of technology [13]. In his performances, the audience can see the direct relationship between the performer's physical state and the musical output. Therefore the audience's perception and interpretation of the music are directly influenced by the performer's physical state and the observation of that state is part of the performance in itself. Donnarumma uses a range of technologies such as electromyography (EMG), electrocardiography (ECG), and respiratory monitoring devices to measure his physiological signals. He also employs signal processing techniques to extract information from raw data and map it to musical parameters. This way, he creates compositions that reflect his physical state at that particular moment. In his performances, Donnarumma explores the relationship between technology and the body and how the use of technology can enhance and extend human physical abilities, more specifically, how technology can shape the human experience and the way one perceives the world around us [13]. His works reside conceptually in the intersection between computational processes and the politics of the body. His body is used as a tool for exploration, creating hybrid and fluid performances that blur the boundaries between human and machine. Fluidity in all its manifestations is very characteristic in these works: the music develops and constantly changes, regardless if the body leads the processes or if technology does. In his work all the elements (body and technology) are presented as a unified [2]system, unaffected by binary distinctions, they are part of the continuous production of space.

3. COMPOSITION / PERFORMANCE

3.1 Composition

The theoretical analysis described here lead the making of the interactive performance "Cyborgutt", composed by sound artist φø (pronounced Faux, also known as Fotis Rovolis) in 2022. Cyborgutt is based on sound material generated by φø's own body. The performance has two main elements; it consists of the live production of a sound environment from the performer's breathing and heartbeat, alongside the simultaneous playback of pre-recorded sound material also deriving from the performer's body. The two sound elements undergo live processing through biosensors placed on the composer's body.

During the composition stage of the fixed sound environment, a series of recordings took place in order to build a sound library. The recordings were divided into three categories based on the recording processes used, the qualities and the characteristics of the sounds: i) external sounds (sounds produced at the surface of the body p.ex. skin friction), ii) internal sounds (sounds produced by the internal organs of the body p.ex. bowel movements), and iii) intermediate sounds (such as voice and breathing). Subsequently, a series of processes took place, the sounds were modified using frequency modulation and granular synthesis, generating new textures, removing gender traces from the voice, aiming to explore new aural textures that are not bound to binary gender definitions and identifiers.

An important role in the recording process was the use of an electronic stethoscope, which was also used later in the live performance of the work. Through the stethoscope, one is able to hear sounds that the naked ear cannot perceive. Looking for the sounds of the internal body became a pivotal part of the composition because it enhanced the overall exploration of the performer's identity through contact with their body and sounds. Further on, based on the collected material, the sound artist created a fixed soundscape by over-stacking three layers of sound events: the rhythms of breathing cycles, the heartbeats, and the stomach and bowel movement sounds. These elements overlap during the composition, with their transitions and rhythms reinforced by the sound material of external sounds.

3.2 Rehearsals

For the live performance of the work, the use of bio-sensors plays a crucial role. A breath sensor, a heart rate sensor, and a muscle sensor linked to a Max/MSP patch are utilised to synchronise samples from the fixed-media soundscape. The breath sensor controls the playback speed of the sound event. The lower and higher values resulting from the lowest and highest points of inhalation and exhalation are mapped to the speed values of the sample. The heart rate sensor controls the duration of the grains. Any anxiety or tension occurring during the performance will alter the breath rate and consequently the heart rate. Therefore, any small

[2] Performances such as "Hyppo Chryssos" (2021)

variation in the range of the incoming values corresponds to a range capable of adjusting the size of the grains in the granular synthesis processes taking place in Max/MSP (figure 2). Finally, the muscle sensor controls the frequency range of the changes occurring in the grain frequency, with each contraction and relaxation being associated with a higher and a lower value of the pitchshift parameter. This layer is closely connected to the two layers of breathing and heart, so they all work together as the living organism of the piece, an organism with its internal processes that respond to the fix-media soundscape.

Rehearsing such a type of work is physically difficult, the performer may find it hard to exceed the duration of 10 minutes due to hyperventilation from the continuous repetitions of inhalation and exhalation. For best results with the breath sensor, one should hold their breath for several seconds while emptying the lungs completely and refilling them fully. Without the necessary preparation, this process may cause dizziness and fatigue in the body. These effects offered the possibility to explore methods for extending the physical endurance of the performer (p.ex. meditation), and yet influenced the final duration of the performance.

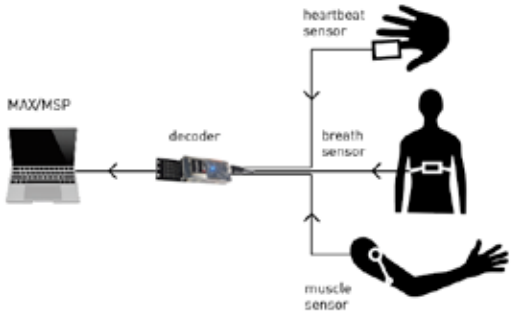


Figure 2. Signal flow

3.3 Performances

In the first performance the performer/sound artist positioned their body obliquely, they had the sensors placed on them, linked to a computer and a controller that were placed right in front of them, next to their head there was a microphone with a stand. The tension increased as the piece progressed, stemming from the combination of the performer's physical and emotional state because of the act of the performance itself as much as because of the hyperventilation. The performer's intention was to take as deep and long breaths as possible to make the sensor reach its extreme-most values.

The second performance (figure 3) was an attempt to make less obvious the relationship between the body and the computer. The aim was to shift the focus entirely on the performer's body. The computer was removed from the stage, and the controller was removed altogether from the process, any changes in the parameters that were previously made by the controller were automated. Likewise, the microphone placed on a stand was replaced with a wireless head microphone, and an electronic stethoscope was added to the system reinforcing the sound of the heartbeat. With the addition of the stethoscope, the total duration of the piece was extended to 14 minutes. This version of the performance had an additional section, a moment where all other sounds went to the background offering space to the sounds of the live and pre-recorded heartbeats; these two elements offered a rhythmic variation to the piece. The changes in this second performance of the piece, although might seem minimal or insignificant, offered a strong impact to the overall experience. The only bodily movement in the second version of the performance was the chest and belly movement caused by the breath: the body therefore gained an even more central role.

The observations that emerged after the presentations span on different levels: from practical to conceptual aspects, creating space for further development, perhaps even a continuous reconfiguration of the project as a whole. A fundamental observation concerns the personal aspect of the work, it is a continuous exposure of the physical and emotional state of their body, giving also an ephemeral aspect to each presentation. What was noticed by the majority of the audience was the strong intensity of the performance. The sharing and presentation of sounds from within the artist's body, revealed an aspect of vulnerability. These sounds generate a sense of familiarity and ease the listeners to surrender and immerse into the flow of the soundscape. Further on regarding the content of the performance and the overall experience, the audience claimed that the work generated a sense of suspension and exploration, it fostered strong variations in emotions fluctuating from anxious moments to moments of calmness suggesting a sense of focus to the present.

In the practical aspect of the work, the observations that emerged relate to the end of the work and its overall structure and duration. Most of the audience members expressed their need for a longer duration of the work in order to become more perceivable (aurally and visually) the impact and the fatigue of the performance processes on the artist's body. Such a longer duration would offer space for a deeper understanding of the concepts dealt by the performance. By significantly increasing the performance duration and by extending the different sections of it, alternating silences with escalation of intensities, and adjusting the structure accordingly to maintain the performer's involvement increasingly, the work would become much more demanding taking on aspects of a long-durational performance. An extended duration of the work may allow experimentation on the endurance of the body, something that might contribute to the overall experience as the artist's complete physical exhaustion would become apparent. Another aspect that emerged as a potential field for exploration was the positioning of the audience. One suggestion was that the artist's body should be placed in the centre with the audience around it. Nonetheless, the work seems to demand exploration on alternative seating options.

Figure 3. “Cyborgutt” performance

4. CONCLUSIONS

The valuable feedback from the audience of both performances (in form of discussions that took place at the end of both events), and the personal reflections of the performer, informed the final conclusions of this paper.

Observing Masumi's analysis of the body and his alternative perspective that moves away from the boundaries of the skin and elevates it to a system of relationships between bodies, time, and space, what becomes obvious, is the relativity of our perception and the fact of a constant variability that governs each body as part of a network. Therefore, the exploration of processes that examine, deconstruct, and potentially redefine traditional notions are important for our liberation from barren boundaries and misconceptions.

It becomes clear that fluidity as a framework for existence offers itself as a tool for understanding and exploration in different contexts. Every body and every sound is unique, their properties might be labeled in one or another way merely because they might operate within specific subjectively defined boundaries. The notion of non-binary identity of body sounds becomes the pivot for the making of an original sound art / performance, challenging conceptions, etiquette and taboos. The use of the electronic stethoscope in the music performance played exactly that fundamental role: it served as a means to challenge these misconceptions by establishing a nuclear connection with the sounds of the body by integrating them live into the sound landscape. Listening to these sounds, which carry rhythms and operate on a contemplative repetitive basis, induces tranquility and immersion in the present moment.

The sensors were crucial both for the visual and the conceptual aspect of the performance. They contributed in limiting the visual noise from the stage by removing all computers and controllers, and at the same time they gave the keys of control of the entire aural experience to the physical and mental capacities of the body, rendering the work much more conceptually integral. The experience of executing a piece without having a rational and physical control over it (by using midi controllers or similar) but having the body and the movements of its vital functions to be in charge of the development of the work makes the experience much stronger and much more personal overall. Throughout the duration of the recordings and ultimately the performance, the body embarks on a feedback loop between the sounds it hears and the sounds it makes, bringing the processes of composition and execution into the discovery of new pathways of musical performance, musical expression and introspection.

The replacement of the conventional computer controllers with sensors that monitor physiological processes of the body, affect eventually the individual's receptivity, as they allow the body to control different sound parameters, more than it is accustomed to, while simultaneously perceiving the sonic outcomes of this process and its internal state. This self-sustaining system approaches the essence of the synthetic sequence attempted in live presentations of cyborg artistic practices. Considering that the characteristics of cyborgs suggest the enhancement of the body capabilities and the production of new sensations, the critical approach implemented in this creative project suggests such a stance to a substantial degree. Following Haddow’s proposition about everyday cyborgs and the suggestions of Harbisson and Ribas regarding the cyborg's characteristic as something independent of augmented bodily capabilities, but rather as a self-identification property, this performance showcases some of these notions and underlines aspects of the cyborg totality.

Regarding the cyborg experience, it is important to remember Stern's observation about the confinement imposed on body and sound, confinement caused by digitality (Stern, 2018), the finite nature of interactive techniques and the use of sensors. These techniques inherently provide a control that parametrises movements, delimiting their outcomes and intensifying them within a predefined set of specific conditions. This trajectory however diverges from the concept of fluidity and deconstruction of predefined perceptions, allowing for some solutions but not leaving room for an infinite range of possibilities that may emerge after each action. Therefore, achieving increment in randomness is becoming a primary objective within the framework of such research paths, a process that could possibly be realised through the integration and experimentation with AI systems.

Last but not least, this performance constitutes an open segment of a continuous journey, still in development. The piece couldn't be considered as an end, given that the journey’s attempt is to transcend finish lines, highlighting the importance for fluidity of every action. The work manages to create a temporal event, a framework of an aural coexistence, in which the audience and the performer explore sounds without limitations and come into contact with the body and the constant alternating sound intensities it creates. On a first level, by observing the fluid dynamics that define the conditions that make the sound work, one may say that the sound universe it creates is of a queer character. In a second instance though and by giving it a closer look, one may argue that the soundscape constitutes a queer substance, due to the content of the sounds from which it consisted. By collecting sounds that are considered taboo either in a musical context or in social conventions (stomach sounds, sounds of pleasure, breath, etc.), as well as sounds from marginal bodies, such as a non-binary body in this case, the composition is constituted as inherently queer.

Future variations could accommodate more

bodies creating a wider sonic fusion, as well as three-dimensional use of sound, given the relationship of space to body and sound. By including the audience in the system of sensors and incorporating multiple bodies at once, the possibility of greater immersion and inclusion of the audience arises. In a similar direction, by extending the time of the performance and pursuing a durational installation, a deeper immersive experience can be achieved. Meanwhile and lastly, expanding the time, the performer's fatigue and physical exhaustion emerges as a potential closure of the work, with the literal extinguishing of the body.

Acknowledgments

We would like to thank Thanos Vovolis for providing references and valuable information and M. Eugenia Demeglio for her help and overall support.

5. REFERENCES

1. J. Butler, *Bodies that matter*. New York: Routledge, 1993
2. J. Halberstam, "What's that smell?," *International Journal of Cultural Studies*, vol. 6, no. 3, pp. 313–333, Sep. 2003. doi:10.1177/13678779030063005
3. F. Jarman-Ivens, Queer Voices Technologies, Vocalities, and the Musical Flaw. New York: Palgrave Macmillan, 2011
4. C. A. Marchiselli, "Queer sonorities: Sound as persuasion in Gertrude Stein's*tender buttons*," *Women's Studies in Communication*, vol. 39, no. 1, pp. 69–85, Jan. 2016. doi:10.1080/07491409.2015.1127302
5. B. Massumi, *Parables for the Virtual: Movement, Affect, Sensation*. Durham: Duke University Press, 2022.
6. N. Stern, *Implicit Bodies Through Explicit Action. Perspectives: International Postgraduate Journal of Philosophy* 1 (1):64-87.
7. G. Thomas "SEX/SEXUALITY & SYLVIA WYNTER'S 'BEYOND...': ANTI-COLONIAL IDEAS IN 'BLACK RADICAL TRADITION'" *Journal of West Indian Literature* 10, no. 1/2 (2001): 92–118. http://www.jstor.org/stable/23019781.
8. D. J. Haraway, *Cyborg Manifesto*. Victoria, British Columbia: Camas Books, 2018.
9. G. Haddow, *Embodiment and Everyday Cyborgs: Technologies That Alter Subjectivity*. Manchester: Manchester University Press, 2021.
10. V. Michielon, "Becoming illegible. Jaime del Val: How to disrupt control society," Digicult, https://digicult.it/news/becoming-illegible-jaime-del-val-how-to-disrupt-control-society/ (accessed Mar. 10, 2024).
11. "What it's like to be a cyborg, as explained by a Cyborg," VICE, https://www.vice.com/en/article/gvv7dj/what-its-like-to-be-a-cyborg-as-explained-by-a-cyborg (accessed Mar. 10, 2024).
12. J. C. Hu, "Sense hacking: The real-life cyborgs of the DIY augmentation scene," How We Get To Next, https://www.howwegettonext.com/sense-hacking-the-real-life-cyborgs-of-the-diy-augmentation-scene/ (accessed Mar. 10, 2024).
13. E. R. Miranda, *Guide to Unconventional Computing for Music*. Cham: Springer International Publishing, 2018.

TRAJECTORY DESCRIPTORS: AN APPROACH TO MUSIC GENRE CLASSIFICATION THROUGH CURVES AND VECTORS IN THE TONNETZ

Christophe WEIS (christophe.weis@stud.hfm.eu) [1],
Marlon SCHUMACHER (schumacher@hfm-karlsruhe.de) [1], and
Moreno ANDREATTA (andreatta@math.unistra.fr) [2]

[1] *Institut fur Musikinformatik und Musikwissenschaft*, **Hochschule fur Musik Karlsruhe**, Karlsruhe, Germany
[2] *CNRS-Institute for Advanced Mathematical Research, ITI CREAA*, **Université de Strasbourg**, Strasbourg, France

ABSTRACT

We present an approach to geometrically represent and analyze the harmonic content of musical compositions based on a formalization of chord sequences as spatial trajectories. This allows us in particular to introduce a toolbox of novel descriptors for automatic music genre classification. Our analysis method first of all implies the definition of harmonic trajectories as curves in a type of geometric pitch class spaces called Tonnetz. We define such curves by representing successive chords appearing in chord progressions as points in the Tonnetz and by connecting consecutive points by geodesic segments. Following a recently established hypothesis that assumes the existence of a narrow link between the musical genre of a work and specific geometric properties of its spatial representation, we introduce a toolbox of descriptors relating to various geometric aspects of the harmonic trajectories. We then assess the appropriateness of these descriptors as a classification tool that we test on compositions belonging to different musical genres. In a further step, we define a representation of transitions between two consecutive chords appearing in a harmonic progression by vectors in the Tonnetz. This allows us to introduce an additional classification method based on this vectorial representation of chord transitions.

1. INTRODUCTION

Using the Tonnetz – that is primarily defined to describe a spatial organization of pitch classes – as the main framework for automatic music genre or music style classification is a task put forth by recent works in Computational Musicology and Music Information Retrieval.

In terms of common formalization and usage, the Tonnetz can be considered to be a two-dimensional triangular grid, whose vertices represent different pitch classes $p \in \mathbb{Z}/12\mathbb{Z}$. The neighboring relationships between these pitch classes are determined by three pitch-class intervals $i_k \in \mathbb{Z}/12\mathbb{Z}$ (for $k \in \{1, 2, 3\}$), that describe the intervallic relations between pitch classes along each of the three axes of the Tonnetz grid. An example of a geometric realization of the Tonnetz is illustrated in Figure 1.

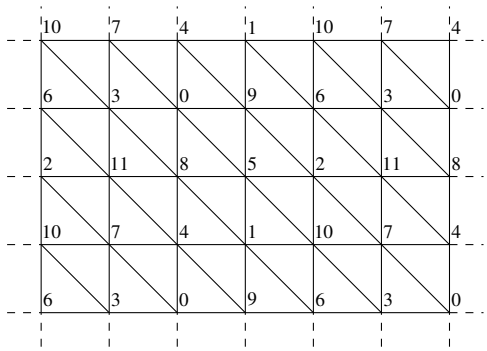


Figure 1. Representation of the Tonnetz based on a two-dimensional triangular grid. The different vertices of the grid represent pitch classes from 0 (or C) to 11 (or B). In this example, the neighboring relationships between the different pitch classes are generated by the pitch class intervals $i_1 = 3$ (along the horizontal axes), $i_2 = 4$ (along the vertical axes) and $i_3 = 5$ (along the oblique axes).

In recent approaches, Bigo et al. [1,2] and Karystinaios et al. [3] introduce formalizations that allow to represent the entire harmonic content of musical pieces by trajectories in the Tonnetz. More precisely, by basing themselves on a vision of the Tonnetz as a simplicial complex, they propose to represent chord progressions by *chord complexes*. These chord complexes are constructed as the union of the pitch classes forming the considered chords and all the edges connecting these pitch classes. Louis Bigo then establishes the hypothesis that there exists a significant link between the style assigned to musical – and especially harmonic – content and certain geometric aspects of the spatial representation of this musical content. In particular, he introduces a measure to assess the compactness of such a representation in the Tonnetz.

Following the research paradigm implying such harmonic trajectories, we present, in this paper, a novel method to spatially represent chord progressions by curves in the Tonnetz. Contrary to the works of Bigo et al. and Karystinaios et al., our mathematical formalism involves describing the Tonnetz as a continuous space by embedding it on a flat torus (section 2.1). This allows us to define a procedure to represent chords by individual points in

Copyright: © 2024. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

this continuous Tonnetz (section 2.2). Relying on this theoretical context, we then construct curves – our Tonnetz *trajectories* – that represent entire chord progressions by gradually connecting the positions in the Tonnetz of successive chords (section 2.3). Once these trajectories have been constructed, we consider a formalism that allows us to represent transitions between two consecutive chords as vectors in the Tonnetz, that we shall call *transition vectors* (section 2.4).

In order to measure various geometric aspects of our Tonnetz trajectories, we introduce a toolbox of several *trajectory descriptors* that are conceived to relate to specific geometric properties and spatial qualities of the trajectories (section 3). We then use both *trajectory descriptors* and *transition vectors* as features for automatic style classification (section 4).

Through our research, we explore a novel way of geometrically representing and analyzing musical harmony. By defining Tonnetz *trajectories*, *transition vectors* and *trajectory descriptors*, we propose novel formalisms and tools developed for analytical tasks in computational music analysis and music theory. By using our tools as classifier features, we demonstrate novel applications in the field of automatic style classification of music.

2. TONNETZ TRAJECTORIES

2.1 Embedding the Tonnetz on a flat torus

The geometric and topological properties of the Tonnetz and its generalizations as well as its suitability as a tool for harmonic analysis have been the subject of numerous mathematical and musicological studies (see e.g. [4–9]). For the purposes of this article, we restrict our approach to the Tonnetz generated by the pitch class intervals 3, 4 and 5 that we initially consider to be a pitch class based, labeled graph. In this graph, the different nodes are labeled by the 12 pitch classes such that each pitch class is represented exactly once and such that the neighbors of each pitch class are exactly the pitch classes that are either a minor third, a major third or a fifth away.

Our actual formalization of the Tonnetz – that, henceforth, we shall denote by $\mathcal{T}(3,4,5)$ – is then based on the observation that this graph is toroidal (see e.g. [5] for an in-depth study of the geometric and topological properties of different Tonnetz generalizations). This shall allow us to naturally embed the graph of the Tonnetz on a flat torus [1]. To be more precise, we propose to consider the rectangular flat torus [2]

$$\mathbb{T} := \mathbb{R}^2 / (4\mathbb{Z} \times 3\mathbb{Z}), \tag{1}$$

together with the canonical projection

$$\pi : \mathbb{R}^2 \to \mathbb{T}. \tag{2}$$

We then define our embedding of the Tonnetz graph on this specific rectangular flat torus by associating, for $i, j \in \mathbb{Z}$, the node labeled by the pitch class $(-3i+4j) \bmod 12$ with the point $\pi(i,j)$ on the flat torus.

[1] (see e.g. [10, 11] for a detailed theoretical framework on flat tori)

[2] An equivalent definition describes this rectangular flat torus as the quotient space of $\mathbb{R}^2$ under the identifications $(x,y) \sim (x+4,y) \sim (x,y+3)$ for all $(x,y) \in \mathbb{R}^2$.

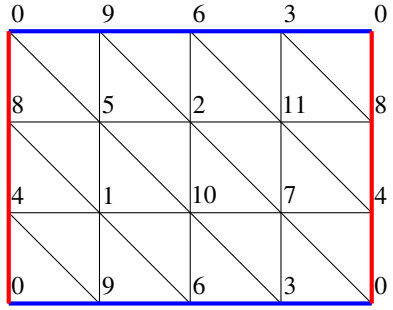


Figure 2. Visualization of the Tonnetz space $\mathcal{T}(3,4,5)$. The embedding of the Tonnetz on a flat torus is characterized by the rectangular *fundamental region* of the flat torus with a pairwise identification of the opposites sides.

As visualized in Fig. 2, the resulting Tonnetz space corresponds to the rectangular *fundamental region* of the underlying flat torus that pairwise identifies its opposites sides. The pitch class 0 (or C) becomes assigned to the point with coordinates $(0,0)$. All horizontal axes represent interval leaps corresponding to the pitch class interval 3, vertical axes correspond to the pitch class interval 4, oblique axes to the pitch class interval 5.

2.2 Placing chords in the Tonnetz

Aiming to use the Tonnetz as a harmonic space, we follow the examples of Bigo et al. [1,2] and Karystinaios et al. [3] and propose to expand the representation potential of the Tonnetz as a pure *pitch class space* to that of a *pitch class set space*. The two approaches mentioned tackle this representation problem by considering chords as being represented in the Tonnetz by chord complexes built up from all the pitch classes defining the chord in question as well as the edges connecting these pitch classes.

In contrast to this approach, we propose a method that represents chords as isolated points (rather than larger complexes) in the Tonnetz. Therefore, we suggest to determine the centroid of all the pitch classes defining a chord.

More precisely, given a chord in the form of a pitch class set $\mathbf{C}$ of size $N \in \{1,\ldots,12\}$, we consider the set of all nodes $\{P_n\}_{n\in\{0,\ldots,N-1\}}$ in the Tonnetz space $\mathcal{T}(3,4,5)$ (as defined in section 2.1 via an embedding on a flat torus) that are labeled by the pitch classes defining $\mathbf{C}$.

In a second step, we *unfold* our flat torus Tonnetz in order to obtain, for $n \in \{0,\ldots,N-1\}$, the preimages $\pi^{-1}(\{P_n\}) \subset \mathbb{R}^2$ under the projection map π of these labeled nodes.

Among these preimages, we then construct all the N-polygons [3] $\mathcal{P}$ of minimum perimeter whose corners are labeled exactly once by the different pitch classes of the initial pitch class set $\mathbf{C}$.

In a last step, we determine the centroids (or geometric centers) $G_{\mathcal{P}}$ of all these polygons of minimum perimeter and place them back into Tonnetz space $\mathcal{T}(3,4,5)$ via

[3] (i.e. polygons with exactly N corners)

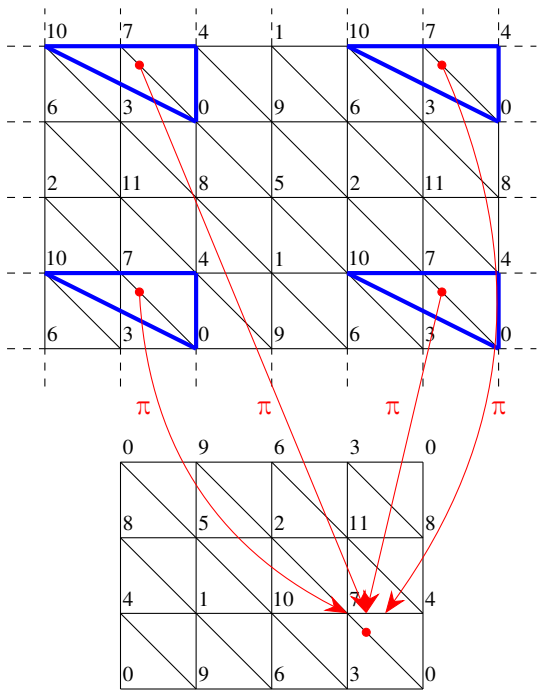


Figure 3. Illustration of the placement procedure for the pitch class set $[0, 4, 7, 10]$ (corresponding to the chord C^7) in the Tonnetz $\mathcal{T}(3, 4, 5)$. We therefore place ourselves in an *unfolded* version of the Tonnetz *(upper part of the figure)*. In this unfolded Tonnetz embedded in $\mathbb{R}^2$, we first determine all the polygons bounded by the pitch classes 0, 4, 7 and 10 that have minimum perimeter *(marked in blue)* and computer their centroids *(marked in red)*. These centroids are then transported back to the original *folded* version of the Tonnetz via the projection map π and finally define the position that represents our given pitch class set $[0, 4, 7, 10]$ in the Tonnetz.

the projection map π.[4] Finally, we choose to define the placement of the pitch class set $\mathbf{C}$ by the set of all points $\pi(G_{\mathcal{P}}) \in \mathcal{T}(3, 4, 5)$ found through this procedure.
In Fig. 3, we illustrate the hereabove described placement procedure for the pitch class set $[0, 4, 7, 10]$, corresponding to the chord C^7.

Having defined a procedure to represent chords (given in the form of pitch class sets) by isolated points in the Tonnetz, we are now set to represent chord progressions[5] by *trajectories* (defined in the following section 2.3) and chord transitions by *vectors* (defined in section 2.4) in our Tonnetz space $\mathcal{T}(3, 4, 5)$.

[4] In this step, it becomes apparent why it is necessary to first *unfold* the Tonnetz space. Indeed, some chords, such as the chord described by the pitch class set $[1, 4, 7, 10]$, allow multiple equally valid positions in the Tonnetz. By unfolding the Tonnetz, line segments connecting the pitch class 4 to the pitch class 7, or connecting the pitch class 1 to the pitch class 4, or connecting the pitch class 10 to the pitch class 1, or finally, connecting the pitch class 7 to the pitch class 10, all describe polygons of minimum perimeter that pass through all the pitch classes of the chord. Thus, these line segments induce four valid positions for the pitch class set $[1, 4, 7, 10]$ in the Tonnetz under the projection map π.

[5] In the following sections, we shall use the term *chord progression* to denote a succession of pitch class sets, the term *chord transition* to denote a chord progression of length 2, i.e. a succession of two directly consecutive pitch class sets.

2.3 Trajectories representing chord progressions

Basing ourselves on the above described procedure to represent chords by points in the Tonnetz, we propose to introduce a method to represent chord progressions by curves. Intuitively, given a chord progression $(\mathbf{C}_0, \ldots, \mathbf{C}_{N-1})$, we therefore trace a piecewise affine curve in the Tonnetz that connects the successive positions associated with the different pitch class sets $\mathbf{C}_n$ (for $n \in \{0, \ldots, N-1\}$).

As a matter of comprehensibility, we here present our construction algorithm for such Tonnetz trajectories by assuming that the first chord of a given chord progression admits only one unique representation, i.e. as a single point in the Tonnetz.[6]

Under this assumption, the construction algorithm for Tonnetz trajectories is iteratively established as follows:

First of all, the position of the initial chord $\mathbf{C}_0$ shall define the starting point $x_0 \in \mathcal{T}(3, 4, 5)$ of the trajectory.

Then, for $n = 1, \ldots, N-1$, it is necessary to distinguish between two cases.

If the chord $\mathbf{C}_n$ is represented by a unique point x_n in the Tonnetz, we newly construct the minimal geodesic segment joining the position $x_{n-1} \in \mathcal{T}(3, 4, 5)$ (which is the position of the previously placed chord $\mathbf{C}_{n-1}$) to the point $x_n \in \mathcal{T}(3, 4, 5)$.

If, however, the chord $\mathbf{C}_n$ admits several positions in the Tonnetz, we propose to arbitrarily privilege the position closest to the point $x_{n-1} \in \mathcal{T}(3, 4, 5)$ (which is, again, the position of the previously placed chord $\mathbf{C}_{n-1}$). If then, still more than one point comes into question, we privilege the position such that the oriented angle between the vector $(1, 0)$ and the velocity vector of the newly constructed geodesic segment becomes minimal.

In any of these cases, we denote the newly constructed minimal geodesic segment by $\gamma_n : [n-1, n] \to \mathcal{T}(3, 4, 5)$.

Our final trajectory shall eventually be given by the concatenation of the $N-1$ found geodesic segments γ_n (for $n = 1, \ldots, N-1$).

By way of example, we show, in Fig. 4, the step by step construction of the trajectory representing the chord progression $\mathrm{Cmaj}-\mathrm{C}^7-\mathrm{Fmaj}-\mathrm{F}\sharp\mathrm{dim}^7$ – or, in terms of pitch class sets, the progression $([0, 4, 7], [0, 4, 7, 10], [0, 5, 9], [0, 3, 6, 10])$.

2.4 Transition vectors representing chord transitions

Based on this trajectory construction procedure, we naturally define a way of representing a transition between two chords by a vector in the Tonnetz. For this purpose, let $(\mathbf{C}_0, \mathbf{C}_1)$ be a chord transition represented by a trajectory consisting of a geodesic segment between two points x_0 and x_1 in the Tonnetz space $\mathcal{T}(3, 4, 5)$. We furthermore consider the parametrization:

$$\gamma : [0, 1] \to \mathcal{T}(3, 4, 5), \tag{3}$$

[6] As seen in section 2.2, some chords, especially chords showing specific symmetries, indeed admit more than one valid position in the Tonnetz. Diminished seventh chords, for instance, are represented by 4 symmetrically distributed points.

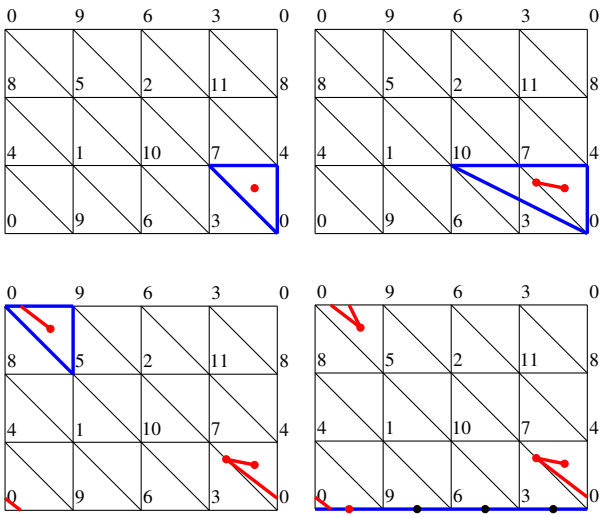


Figure 4. Step by step construction of the trajectory representing the chord progression $([0,4,7],\ [0,4,7,10],\ [0,5,9],\ [0,3,6,10])$ The beginning of the trajectory is given by the minimal geodesic segment connecting the positions of the chords $[0,4,7]$ and $[0,4,7,10]$ *(upper right part of the figure)*. We then determine the minimal geodesic segment between the positions of the chords $[0,4,7,10]$ and $[0,5,9]$, that visually passes through the four sides of the rectangular fundamental region of the Tonnetz space *(lower left part of the figure)*. The last step requires determining the point closest to the chord $[0,5,9]$ among the four points representing the chord $[0,3,6,9]$ in the Tonnetz *(lower right part of the figure)*.

with $\gamma(0) = x_0$ and $\gamma(1) = x_1$ of this geodesic segment. Then, we define the *transition vector* associated with the chord transition $(\mathbf{C}_0, \mathbf{C}_1)$ by the velocity vector:

$$\gamma'(0) \in T_{x_0}\mathcal{T}(3,4,5) \cong \mathbb{R}^2, \tag{4}$$

where $T_{x_0}\mathcal{T}(3,4,5)$ denotes the tangent space of $\mathcal{T}(3,4,5)$ in the point x_0.

3. TRAJECTORY DESCRIPTORS

As an essential part of the framework of our research, we globally follow the paradigm – significantly put forth by the works of Bigo et al. [1] and by Karystinaios et al. [3] – that presumes the existence of a narrow link between the stylistic nature assigned to the harmonic content of a musical piece and specific geometric characteristics of its corresponding spatial representation. In order to measure or capture various properties of our Tonnetz trajectories, we introduce a toolbox of several descriptors that we specifically designed to describe geometric aspects of these trajectories and that can potentially be related to harmonic characteristics of the represented chord progressions.

In Table 1, we give an overview of all the trajectory descriptors introduced below.

Later, in section 4, we propose an application towards testing these trajectory descriptors as features implied in an automatic classification tool for chord progressions.

In the following sections, we assume that a Tonnetz trajectory of length N, i.e. a trajectory representing a chord progression of length N, is given by the parametrization:

$$\Gamma:\ [0,\ N-1]\ \rightarrow\ \mathcal{T}(3,4,5), \tag{5}$$

where, for $n = 0, \ldots, N-1$, the point $\Gamma(n)$ corresponds to the position associated to the trajectory's n^{th} chord in the Tonnetz.

3.1 The number of different transition vectors

From the definition of transition vectors introduced in section 2.4, it follows intuitively that chord transitions of the same nature are represented by vectors of the same length and direction.[7] In particular, this entails that any succession of specific scale degrees (e.g. the progression $\mathbf{II}-\mathbf{V^7}-\mathbf{vi}$ in a major scale) is represented by the same succession of transition vectors when being transposed.

This simple observation leads us to introduce a first trajectory descriptor that counts the number of different transition vectors that appear within a given trajectory. This shall allow us to obtain a first estimation on the variety of chord transitions used in a chord progression and, thereby, an estimation on the richness of the harmonic language in use.

3.2 The number of different vector orientations

As a descriptor that is closely linked to our first descriptor (counting the number of different transition vectors in a trajectory), we propose to additionally take count of the number of different vector orientations. Practically, we therefore consider collinear vectors as being part of a same *collinearity class* and by counting the number of *collinearity classes*. Geometrically, this boils down to counting the number of axes in the Tonnetz along which a given trajectory evolves.

3.3 The number of triangular regions visited by a trajectory

The Tonnetz $\mathcal{T}(3,4,5)$ being generated by interval classes corresponding to the minor and major thirds as well as the fifth, it naturally turns out to be a geometric space that is well-suited for representing major and minor triads. Indeed, it is obviously apparent that the pitch classes defining major and minor chords delimit triangular regions inside the Tonnetz space $\mathcal{T}(3,4,5)$. In order to estimate the tonal mobility of a given chord progression, we propose to take advantage of this property and count the number among these triangular regions that the trajectory of this chord progression passes through. We particularly reckon that highly modulation progressions, by visiting a high number of tonalities, imply trajectories that traverse a high number of triangle regions.

[7] The formal framework presented in this paper does not, however, allow to establish a one-to-one correspondence between the set of possible transition vectors and the set of all possible chord transition natures. In other terms, two vectors of the same length and direction may represent two chord transitions of different nature. A more refined representation method, for instance by assigning different heights to different chord types, may help remedy this issue, but is not required in the context of the basic definitions of our trajectory descriptors.

3.4 Distances between chords and angles between transition vectors

Still pursuing the idea of estimating characteristic aspects of the tonal path of a composition, we take into account the average distance between two successive chords. More formally, given the parametrization of a Tonnetz trajectory as defined by (5), the average distance between two successive chords can be given by the formula:

$$\overline{\text{distances}}(\Gamma) := \frac{1}{N} \sum_{n=0}^{N-1} d_{\mathcal{T}(3,4,5)}\Big(\Gamma(n),\, \Gamma(n{+}1)\Big), \quad (6)$$

where $d_{\mathcal{T}(3,4,5)}$ denotes a distance function on the Tonnetz space $\mathcal{T}(3,4,5)$.

As a further descriptor that estimates the average evolution of a trajectory, we propose to also assess the angles between consecutive transition vectors. Formally, we therefore determine the absolute values of the cosine similarity between two consecutive transition vectors. [8] Our descriptor is then given by the average of these values along an entire trajectory i.e., considering again the parametrization (5), we obtain:

$$\overline{\text{cos similarities}}(\Gamma) := \frac{1}{N-1} \sum_{n=0}^{N-2} \left| \frac{\mathbf{v}_n \cdot \mathbf{v}_{n+1}}{\|\mathbf{v}_n\| \, \|\mathbf{v}_{n+1}\|} \right|, \quad (7)$$

where, for $n = 0, \ldots, N-2$, $\mathbf{v}_n$ and $\mathbf{v}_{n+1}$ denote the transition vectors originating from the nth and the $(n+1)^{\text{th}}$ chord respectively and where $\mathbf{v}_n \cdot \mathbf{v}_{n+1}$ denotes the standard Euclidean dot product between these two vectors.

Having introduced these two descriptors, we additionally propose to include in our descriptor toolbox the standard deviation associated with these latter average values.

3.5 The width of a trajectory

In order the get a certain impression of the *width* of a trajectory (starting with chord $\Gamma(0)$ and ending with chord $\Gamma(n-1)$ according to the parametrization (5)), we propose to compare it to a straight line connecting $\Gamma(0)$ with $\Gamma(n-1)$. As such a comparison turns out to be not trivial in a quotient space such as our flat torus Tonnetz space, we propose again to *unfold* our Tonnetz and, thereby, also unfold our given trajectory.

Similar to the unfolding done in section 2.2, we therefore consider the preimage under our canonical projection π of the given trajectory $\Gamma([0,\, N-1])$. The obtained preimage $\pi^{-1}\Big(\Gamma([0,\, N{-}1])\Big)$ shall then correspond to unfolded and connected versions of the initial trajectory that are now embedded in $\mathbb{R}^2$. Among these preimage trajectories embedded in $\mathbb{R}^2$, it shall be sufficient for our purposes to select, as a representative, an arbitrary trajectory that we denote by $\widetilde{\Gamma}$. Through this unfolded version of our trajectory, we finally, assess its width by computing the Hausdorff distance between the curve $\widetilde{\Gamma}([0,\, N-1])$ and the straight line connecting its starting and end points $\widetilde{\Gamma}(0)$ and $\widetilde{\Gamma}(N-1)$.

[8] This shall give us values closer to 0 the closer two vectors are to being orthogonal, values closer to 1 the closer two vectors are to being collinear.

More precisely, we denote by σ the straight line passing through $\widetilde{\Gamma}(0)$ and $\widetilde{\Gamma}(N-1)$ in $\mathbb{R}^2$ and by Σ^+ and Σ^- the half-planes of $\mathbb{R}^2$ determined by σ. Then, our width descriptor shall be defined by the formula:

$$\text{width}(\Gamma) := d_{\text{H}}(\widetilde{\Gamma} \cap \Sigma^+,\, \sigma) \;+\; d_{\text{H}}(\widetilde{\Gamma} \cap \Sigma^-,\, \sigma), \quad (8)$$

where d_{H} denotes the Hausdorff distance in $\mathbb{R}^2$.

3.6 The inconstancy of a trajectory

Our final descriptor is largely inspired by the work of Jean-Paul Allouche and Laurence Maillard-Teyssier [12] on the *inconstancy* of sequences and curves. The idea that we adopt in the context of our paper is to assess the global aspect of a given curve in $\mathbb{R}^2$ in terms of its fluctuations or its curviness, i.e. the more a curve looks irregular, the higher its *inconstancy* should be, whereas the more it resembles to a straight line, the lower its *inconstancy* should be (and eventually tend to 1).

As the inconstancy formula presented by Jean-Paul Allouche and Laurence Maillard-Teyssier implies curves embedded in the Euclidean plane, we shall again, as described in the previous section , unfold any given Tonnetz trajectory Γ in order to obtain a corresponding unfolded trajectory $\widetilde{\Gamma}$ embedded in $\mathbb{R}^2$.

We then can express, according to the theoretical framework described in [12], the inconstancy of the Tonnetz trajectory Γ by the formula:

$$\text{inconstancy}(\Gamma) := \frac{2 \cdot \text{len}(\widetilde{\Gamma})}{\text{per}\big(\text{conv}(\widetilde{\Gamma})\big)}, \quad (9)$$

where $\text{len}(\widetilde{\Gamma})$ denotes the length of the curve $\widetilde{\Gamma}$ and $\text{per}\big(\text{conv}(\widetilde{\Gamma})\big)$ the perimeter of its convex hull in $\mathbb{R}^2$.
We introduce this descriptor particularly to verify its adequateness to keep track of the potential tonal unsteadiness and harmonic richness of given chord sequences. As illustrated by Fig. 5, our inconstancy descriptor specifically returns the value 1 for a sequence of descending fifths, while it returns a higher value for a sequence that moves more irregularly through different chords and/or tonalities.

4. APPLICATIONS TO STYLE CLASSIFICATION

In order to assess the suitability of our trajectory formalism and specifically the trajectory descriptors introduced in section 3 as a tool for automatic style classification, we propose two applications – using, first of all, trajectory descriptors and, in a second step, transition vectors as features for classification.

4.1 Data sets, chord extraction and preprocessing

Our method implies the use of MIDI file data sets that provide pieces belonging to a range of different musical styles from which we retrieve the harmonic content in order to compute their corresponding trajectories in the Tonnetz. For all analyses performed within the context of this paper, we use MIDI files belonging to the MuseData corpus [13] and the Nottingham database [14]. We collected

Descriptor name	*Explanation*
$\lvert\{\text{transition vectors}\}\rvert$	Counts the number of different transition vectors that appear within a given trajectory.
$\lvert\{\text{vector orientations}\}\rvert$	Considers collinear vectors as being part of a same *collinearity class* and counts the number of different *collinearity classes* within a trajectory.
$\lvert\{\text{visited triangular regions}\}\rvert$	Counts the number of triangular regions of the Tonnetz grid that a trajectory passes through.
$\overline{\text{distances}}$	Computes the average distance between two successive chords.
std(distances)	Computes the standard deviation of all the distances between two successive chords.
$\overline{\text{cos similarities}}$	Computes the average cosine similarity between two successive transition vectors.
std(cos similarities)	Computes the standard deviation of all the cosine similarities between two successive transition vectors.
width	Computes the width of a trajectory by considering the Hausdorff distance between the *unfolded* version of the trajectory and the straight line connecting its starting and end points.
inconstancy	Estimates the global aspect of a trajectory in terms of its fluctuations or its unsteadiness.

Table 1. Overview of all the trajectory descriptors defined in section 3.

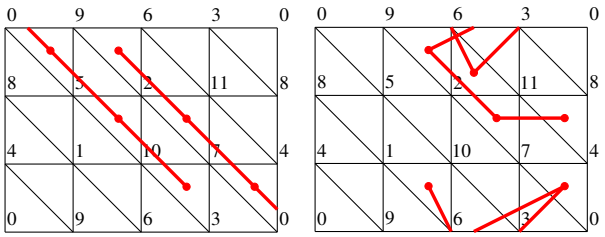


Figure 5. Comparison between the trajectories of the chord progression Dmaj – Gmaj – Cmaj – Fmaj – B♭maj – E♭maj – correspoding to a sequence of descending fifths – *(left side of the figure)* and the chord progression F♯maj – Bmin – Cmaj – Dmaj – Gmaj – Emaj *(right side of the figure)* in terms of their inconstancies. While the first, very regular progression has an inconstancy of 1.0, the second progression, seeming more irregularly fluctuating, has an inconstancy of about 2.17.

284 pieces in total from the MuseData corpus (including chorale harmonizations by J. S. Bach, Trio Sonatas by A. Corelli and various pieces by W. A. Mozart) and 287 pieces from the Nottingham database (consisting of a large collection of British and American folk tunes). Throughout the following sections, these four stylistic contexts shall define our target classes (referred to as *Corelli*, *Bach*, *Mozart* and *Nottingham*) for all classification tasks undertaken.

For each of the MIDI files in our data set, we extract the harmonic content in the form of sequences of pitch class sets using the Python library *music21* [15]. Then, for each of theses sequences of pitch class sets (or, returning to the term used above in this paper, for each chord progression), we determine its Tonnetz trajectory. Subsequently, we consider all the values measured using our trajectory descriptors as well as all the transition vectors implied by the chord transitions represented by the trajectory.

4.2 Trajectory descriptor based classification

In order to implement a trajectory descriptor based classification method, we propose to base ourselves on the k-nearest neighbors (k-NN) algorithm that we use in the style of binary and multiclass classifiers.

Therefore, we first perform measures using our descriptors on the trajectories of randomly chosen pieces belonging to two or more different styles represented in our database. As we are dealing with descriptors that might be impacted by the lengths of the analyzed trajectories, we propose, at this stage of our method, to randomly select trajectory extracts of the same length N to eliminate any length-related bias. More precisely, each of the pieces of our database shall be represented by exactly one trajectory of length N that is selected from a random starting position within the considered piece. Throughout all the classification tests described here below, we propose to set the length of each of the computed trajectories to $N = 20$.

In a next step, we apply the different descriptors defined in section 3 to each of these extracted trajectories and normalize the obtained values with respect to their mean and standard deviation.

In a final step, we use the k-NN algorithm to assess the adequateness of our trajectory descriptors as features for binary and multiclass classification. For this purpose, we choose, for each series of tests, two to four classes present in our data set (e.g. Bach chorales and folk tunes from the Nottingham database for a series of binary classification tests). We then split the chosen data into a training and a test set such that the test set contains exactly 10 pieces (represented by their respective trajectory extracts). We eventually train our k-NN classifier on the training set and evaluate its predictions with respect to the test set using the *accuracy* metric implemented by the formula:

$$\text{accuracy} := \frac{\text{number of correctly classified pieces}}{\text{total number of pieces}}. \quad (10)$$

In Table 2 and Table 3, we give an overview of all the re-

Styles to classify		*Avg. accuracy*
Corelli	Bach	0.80
Corelli	Mozart	0.82
Corelli	Nottingham	0.95
Bach	Mozart	0.84
Bach	Nottingham	0.94
Mozart	Nottingham	0.92

Table 2. Results obtained by using trajectory descriptors as features for binary classification.

Styles to classify	*Avg. accuracy*
Corelli, Bach, Mozart	0.69
Corelli, Bach, Nottingham	0.73
Corelli, Mozart, Nottingham	0.73
Bach, Mozart, Nottingham	0.75
Corelli, Bach, Mozart, Nottingham	0.67

Table 3. Results obtained by using trajectory descriptors as features for multiclass classification.

sults obtained throughout our descriptor based binary and multiclass classification tests.

All the tests in the series referred to in Table 2 and Table 3 were performed by using values obtained using six of our total of nine trajectory descriptors: the *number of different transition vectors*, the *number of different vector orientations*, the *number of visited triangular regions*, the *average distance between successive chords*, the *standard deviation of all these distances* and the *inconstancy*. We justify this selection by our observation that this specific set of descriptors allowed us to achieve the most consistent results over all our test series.

For each of the considered classes, we repeated our testing procedure 10 times by randomly resplitting the respective data sets each time. We provide, in Table 2 and Table 3, the average value of all the accuracy scores obtained across each of these series of 10 tests.

4.3 Transition vector based classification

As a second classification method explored during our research, we propose to rely exclusively on the nature of the different transition vectors retrieved from the harmonic content in our data sets. Thereby, instead of relying on global characteristics of geometric representations as measured by our trajectory descriptors, we focus, in this section, on local aspects of Tonnetz trajectories as expressed by individual transition vectors. From a music theoretical point of view, we base this approach on the assumption that the harmonic style of a corpus becomes implicitly defined through its harmonic content and, specifically through its individual chord transitions. Moreover, we presuppose that two music corpora relating to two different harmonic styles can be distinguished through specific characteristic chord transitions that are contained in one corpus and that are not contained in the other corpus.

To reconnect with the formalism of our transition vectors,

Styles to classify		*Avg. accuracy*
Corelli	Bach	0.97
Corelli	Mozart	0.95
Corelli	Nottingham	1.00
Bach	Mozart	0.96
Bach	Nottingham	0.98
Mozart	Nottingham	0.96

Table 4. Results obtained by using transition vectors as features for binary classification.

given two musical corpora containing pieces belonging to two distinguishable harmonic styles $\mathfrak{C}_1$ and $\mathfrak{C}_2$, we determine the Tonnetz trajectories of all the pieces contained in these two corpora and, subsequently, all the corresponding transition vectors. We denote the set of all transition vectors related to the corpus $\mathfrak{C}_1$ by $\mathfrak{V}_1$, the set of all transition vectors related to the corpus $\mathfrak{C}_2$ by $\mathfrak{V}_2$.

We then propose to introduce the set of transition vectors that are exclusively found in corpus $\mathfrak{C}_1$, and that do not occur at all in corpus $\mathfrak{C}_2$, that we denote by

$$\mathfrak{V}_{1\setminus 2} := \mathfrak{V}_1 \setminus \mathfrak{V}_2 \tag{11}$$

and, similarly, the set of transition vectors that are exclusively found in corpus $\mathfrak{C}_2$, and not present in corpus $\mathfrak{C}_1$:

$$\mathfrak{V}_{2\setminus 1} := \mathfrak{V}_2 \setminus \mathfrak{V}_1. \tag{12}$$

Our classification method can now be described as follows. Given an unknown [9] piece, we determine its Tonnetz trajectory and all of its corresponding transition vectors. If then, among these transition vectors, there are more vectors that can also be found in $\mathfrak{V}_{1\setminus 2}$, we assign the unknown piece to the harmonic style of corpus $\mathfrak{C}_1$. Otherwise, if there are more vectors that can also be found in $\mathfrak{V}_{2\setminus 1}$, we assign it to the harmonic style of corpus $\mathfrak{C}_2$.

In Table 4, we give an overview on several results obtained throughout our vector-related classification tests. The tests in this series were performed using different couples of data sets and by repeating the classification procedure 10 times using randomly selected pieces out of each data set and determining the average accuracy score obtained.

4.4 Comparison between descriptor based and vector based classification

Contrary to the use of trajectory descriptors as classifier features, the current design of our method for transition vector based classification is only optimized for binary classification. In order to compare both types of applications, we summarize the results obtained throughout all our binary classification test series in Table 2 (descriptor based classification) and Table 4 (vector based classification). Overall, the results obtained using transition vectors appear to be systematically more accurate, with only the test series involving the Nottingham class leading to comparable results.

[9] The term *unknown* would, in this context, signify that it is not known whether the piece would belong to corpus $\mathfrak{C}_1$ or corpus $\mathfrak{C}_2$.

On a more technical level, our vector based method requires a particularly careful compilation of the used data sets. In fact, the proposed procedure seems to be highly sensitive to the total amount of different transition vectors contained in the trajectories retrieved from the considered classes. We particularly notice that the use of imbalanced classes (in terms of numbers of different transition vectors) leads to considerably less accurate results.

5. CONCLUSIONS

We presented a formalization allowing to represent chord progressions by curves – the *trajectories* – and chord transitions by vectors – the *transition vectors* – in the Tonnetz. Based on this mathematical framework, we introduced a toolbox of novel *trajectory descriptors* relating to several specific geometric aspects of Tonnetz trajectories. We have shown that using trajectory descriptors and transition vectors have yielded promising results when being used as features in frameworks designed for automatic harmonic style classification. Therefore, future work should aim for a more comprehensive evaluation of the proposed descriptors, for example by using a larger number of styles. Moreover, we emphasize that, by definition, our Tonnetz trajectories can only account for a subset of all the aspects that could make up the richness of a harmonic language. Thus, future work may also imply a larger amount of harmonic parameters, such as chord positions, voice-leading, or harmonic rhythm, as well as other musical parameters in general, in particular rhythm and timbre. This may demand a more profound work on richer, higher-dimensional musical spaces and, eventually, a re-formalization of the concept of trajectories.

Specifically with regard to vectors, a further analysis might involve a reflection on a connection between the transition vectors presented in this paper with the theory of *harmonic vectors* by Nicolas Meeùs [16], which has already been applied in the context of stylistic analyses by Philippe Cathé [17].

While the testing results found throughout our research seem interesting, we also reckon that the formal framework on Tonnetz trajectories might have applications outside of a classification context. For instance, in an educational environment, tracing trajectories in the Tonnetz might become a useful learning or creative tool when implemented in a way that allows to draw curves by hand on a Tonnetz interface and, through this, to freely create new chord progressions.

Acknowledgments

We thank Isabelle Bloch and Carlos Agón for their support and discussions during an initial phase of our research.

6. REFERENCES

[1] L. Bigo, "Représentations symboliques musicales et calcul spatial," Ph.D. dissertation, Paris Est, 2013.

[2] L. Bigo, M. Andreatta, J.-L. Giavitto, O. Michel, and A. Spicher, "Computation and Visualization of Musical Structures in Chord-Based Simplicial Complexes," in *International Conference on Mathematics and Computation in Music*. Springer, 2013, pp. 38–51.

[3] E. Karystinaios, C. Guichaoua, M. Andreatta, L. Bigo, and I. Bloch, "Music Genre Descriptor for Classification Based on Tonnetz Trajectories," *Journées d'Informatique Musicale*, 2020.

[4] D. Tymoczko, "The Generalized Tonnetz," *Journal of Music Theory*, pp. 1–52, 2012.

[5] M. J. Catanzaro, "Generalized Tonnetze," *Journal of Mathematics and Music*, vol. 5, no. 2, pp. 117–139, 2011.

[6] R. Cohn, "Neo-Riemannian Operations, Parsimonious Trichords, and their "Tonnetz" Representations," *Journal of Music Theory*, vol. 41, no. 1, pp. 1–66, 1997.

[7] E. Gollin, "Some Aspects of Three-Dimensional "Tonnetze"," *Journal of Music Theory*, pp. 195–206, 1998.

[8] J. Yust, "Geometric Generalizations of the Tonnetz and their Relation to Fourier Phase Spaces," *Mathematical Music Theory: Algebraic, Combinatorial, Topological and Applied Approaches to Understanding Musical Phenomena*, pp. 253–278, 2018.

[9] G. Baroin and H. Seress, "The Spinnen-Tonnetz: New Musical Dimensions in the 2D Network for Tonal Music Analysis," in *MCM 2015, 5th nternational Conference on Mathematics and Computation in Music*, 2015.

[10] S. I. R. Costa, M. Muniz, E. Agustini, and R. Palazzo, "Graphs, Tessellations, and Perfect Codes on Flat Tori," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2363–2377, 2004.

[11] R. E. Schwartz, *Mostly Surfaces*. American Mathematical Soc., 2011, vol. 60.

[12] J.-P. Allouche and L. Maillard-Teyssier, "Inconstancy of Finite and Infinite Sequences," *Theoretical Computer Science*, vol. 412, no. 22, pp. 2268–2281, 2011.

[13] C. for Computer Assisted Research in the Humanities at Stanford University, "MuseData," http://musedata.stanford.edu (Accessed: 10 March 2024).

[14] E. Foxley, "Nottingham Database," https://ifdo.ca/~seymour/nottingham/nottingham.html (Accessed: 10 March 2024).

[15] M. S. Cuthbert and C. Ariza, "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *International Society for Music Information Retrieval*, 2010.

[16] N. Meeùs, "Vecteurs harmoniques," *Musurgia*, pp. 7–34, 2003.

[17] P. Cathé, "Harmonic Vectors and Stylistic Analysis: A Computer-Aided Analysis of the First Movement of Brahms' String Quartet Op. 51-1," *Journal of Mathematics and Music*, vol. 4, no. 2, pp. 107–119, 2010.

AWAR-

DED

AWARDS SMC 2024

BEST PAPER
BEST MUSIC & SONIC ART

PA

PERS

BEST PAPER

SIMULATING PIANO PERFORMANCE MISTAKES FOR MUSIC LEARNING

Alia Morsi
Universitat Pompeu Fabra
Huan Zhang
Queen Mary University of London
Akira Maezawa
Yamaha Corporation
Simon Dixon
Queen Mary University of London
Xavier Serra
Universitat Pompeu Fabra

BEST MUSIC & SONIC ART

TRANSVARIATIONS – MUSIC BEYOND THE LIMITS OF TIME AND TECHNOLOGY

Oeyvind Brandtsegg
Norwegian University of Science and Technology
Alfonso Benetti
University of Aveiro
Trond Engum
Norwegian University of Science and Technology
Francisco Monteiro
Instituto Politécnico do Porto

PARTICIPANTS

AUTHORS
ARTISTS
ORGANIZERS
VOLUNTEERS

PAR-
TICI-
PANTS

Ahmet Emin Memis Norway
Akira Maezawa Japan
Akira Takaoka, J. F. Oberlin University Japan
Alain Bonardi France
Alberto Barrera Herrero Spain
Alberto Gatti Italy
Alessandro Russo Italy
Alex Chechile USA
Alexander Carôt Germany
Alexander Refsum Jensenius Norway
Alexandre D'Hooge France
Alexandros Kontogeorgakopoulos Greece
Alfonso Benetti Portugal
Alia Morsi Spain
Alo Allik United Kingdom
Álvaro Elena USA
Anastasia Georgaki Greece
Andre Holzapfel Sweden
Andrea Franceschini Italy
Andrea Poltronieri Italy
Andrei Yeudakimau Poland
Andrew Horner Hong Kong
Ângela da Ponte Portugal
Anna Zuccante Italy
Anna-Maria Christodoulou Norway
Annie Mahtani United Kingdom
Arshia Samsaminia Greece
Arthur Jinyue Guo Norway
Binet Anais France
Bruno Pereira Portugal
Çağrı Erdem Norway
Carlos Guedes United Arab Emirates
Carlos Lopes Portugal
Carolina Anahory Portugal
Cárthach Ó Nuanáin United Kingdom
Cat Hope Australia
Christophe Weis Germany
Dan Overholt Denmark
Daniel Hug Switzerland
Daniel Mayer Austria
Daniel Overholt Denmark
David Pirrò Austria
Davide Andrea Mauro Germany

Davide Fantini Italy
Davide Rocchesso Italy
Dermot Furlong United Kingdom
Diamantino Freitas Portugal
Dimitri Papageorgiou Greece
Dimitris Andrikopoulos Portugal
Dimitris Maronidis Greece
Doga Cavdir Denmark
Dr Chris Rhodes United Kingdom
Eduardo Mota Portugal
Eleonora Oreggia United Kingdom
Emanuel Sousa Portugal
Emma Margetson United Kingdom
Enrico Dorigatti United Kingdom
Erika Rumbold Canada
Esteban Gutiérrez Spain
Eugénio Amorim Portugal
F. Amilcar Cardoso Coimbra
Fani Kosona Greece
Fatemeh Alizadeh Germany
Fátima Fonte United Kingdom
Federico Avanzini Italy
Federico Simonetta Italy
Filipa Magalhães Portugal
Filipe Lopes Portugal
Florence Leve France
Florian Colombo Switzerland
Fotis Rovolis Greece
Francesco Foscarin Austria
Francesco Ganis Denmark
Francesco Perissi Italy
Francisca Gonçalves Portugal
Francisco Braga Portugal
Frederico Pereira Portugal
Georg Essl USA
Georgios Marentakis Norway
Gerard Roma United Kingdom
Giada Zuccolo Italy
Gilberto Bernardes Portugal
Giorgio Gnecco Italy
Giorgio Presti Italy
Giovanni Santini United Kingdom
Xi'an Jiaotong United Kingdom

Grigore Burloiu Romania
Guilherme Campos Portugal
Hanna Järveläinen Switzerland
Hans Lindetorp Sweden
Haruhiro Katayose Japan
Henrique Ferreira Portugal
Henrique Portovedo Portugal
Hiroki Nishino Japan
Hung, Tzu-Ching Taiwan
Inês Salselas Portugal
Jackson Loth United Kingdom
Jaime Reis Portugal
Jean-Louis Giavitto France
Jean-Michael Celerier Canada
Jesper Andersen Denmark
Ji Youn Kang Austria
Jingxuan Yue Netherlands
Ji-Xuan Huang Taiwan
João Franco Portugal
João Svidzinski France
Johan Pauwels United Kingdom
Jorge Alves Portugal
Jorge Forero Portugal
Jorge Ramos Portugal
Jorge Sousa Portugal
José Vieira Portugal
Jung In Jung United Kingdom
Juraj Kojs USA
Kaitlin Pet USA
Kit Armstrong Taiwan
Kjetil Falkenberg Sweden
Konstantinos Vasilakos China
Lemon Guo USA
Leo Fogadic Denmark
Leonardo Fierro Finland
Leonie Strecker Austria
Lorenda Ramou Greece
Lorenzo Ballerini Italy
Luc Döbereiner Austria
Luca Andrea Ludovico Italy
Luís Bittencourt Portugal
Luís Costa Greece
Luís Neto da Costa Greece

Luís Rodrigues Portugal
Luna Valentim USA
Maham Riaz Norway
Marcelo Queiroz Brazil
Marco Conceição Portugal
Marco Fiorini France
Margarida Leal Portugal
Martin Ljungdahl Eriksson Sweden
Masatoshi Hamanaka Japan
Matteo Olivo France
Matthew Davies USA
Maxime Popoff France
Michael Boyd USA
Michael Mulshine United Kingdom
Michalis Paraskakis Greece
Michelle Lou USA
Miguel Carvalhais Portugal
Miguel Lourtie Portugal
Mirjam Visscher Netherlands
Moritz Huegel Germany
Nádia Carvalho Portugal
Natália Santos Portugal
Natasha Barrett United Kingdom
Niccolo' Pretto Italy
Nicola Davanzo Italy
Nolan Lem Sweden
Notto Thelle Norway
Nuno Ferreira Ribas Portugal
Nuno Fonseca Portugal
Nuno Peixoto de Pinho Portugal
Olivier Lartillot Norway
Orestis Toufektsis Austria
Øystein Fjeldbo Norway
Øyvind Brandtsegg Norway
Ozan Baysal, Turkey
Panayiotis Kokoras USA
Pedro Louro Portugal
Pedro Pablo Lucas Bravo Norway
Pedro Pestana Portugal
Pedro Rebelo United Kingdom
Pedro Santos Portugal
Pedro Sarmento United Kingdom
Penelope Bekiari Greece

Pierluigi Bontempi Italy
Pierre Alexandre Tremblay United Kingdom
Pierre Cochard France
Pierre Jouvelot France
Poti Martin USA
Pushkar Jajoria Germany
Raul Masu Hong Kong
Recep Gül Turkey
Ricardo Pinto Portugal
Riccardo Russo Italy
Riccardo Simionato Norway
Roderick Buchanan-Dunlop United Kingdom
Rodolfo Ocampo Australia
Roger Dannenberg USA
Romain Michon France
Rory Sweeney Ireland
Roser Batlle Roca Spain
Ruby Crocker United Kingdom
Rui Dias Portugal
Rui Paiva Portugal
Rui Penha Portugal
Rumi Hiraga Japan
Salvador Miranda Portugal
Savvas Kazazis Greece
Scott Wilson United Kingdom
Sean Luke USA
Sebastian Schlecht Finland
Sepideh Shafiei USA
Shapour Hakam USA
Shiguang Liu China
Sofia Vieira Portugal
Staas de Jong Netherlands
Stefania Serafin Denmark
Stefano Catena United Kingdom
Stefano D'Angelo Italy
Stefano Fasciani Norway
Sylvain Marchand France
Takayuki Nakatsuka Japan
Tanguy Risset France
Ted Moore USA
Tejaswinee Kelkar Norway
Telmo Marques Portugal
Teresa Carrasco Garcia Switzerland

Tetsuro Kitahara Japan
Thales Rhoel Pessanha Portugal
Thibaut Carpentier France
Thomas Deacon United Kingdom
Thomas Rushton France
Tilemachos Moussas Greece
Tim Roth, Canada
Tom Barker United Kingdom
Vanessa Faschi Italy
Vicente Lora Portugal
William Walker Smith Netherlands
Xavier Lizarraga Seijas Spain
Xavier Riley United Kingdom
Xiaolong Ma China
Xiaowan Yi United Kingdom
Yenan Zhang Japan
Yi-Wen Liu Taiwan
Yota Morimoto Japan
Youngjae Cho Germany
Yusuf Can Seftali Turkey
Yutian Hu United Kingdom
Zijing Cao Portugal

PHO-

TO-

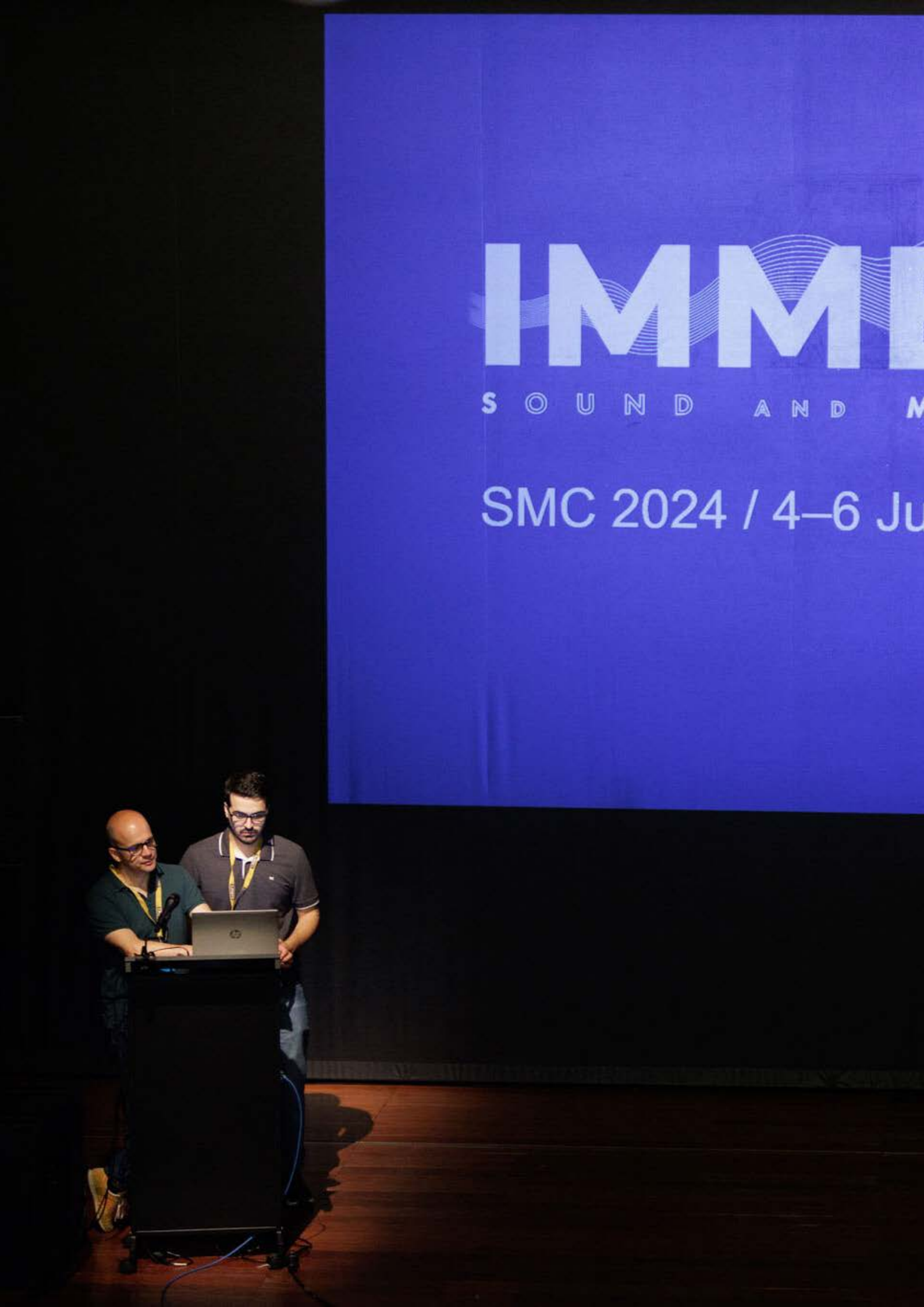
SMC IN PHOTOS

PHOTOGRAPHIC REGIST

GRA-

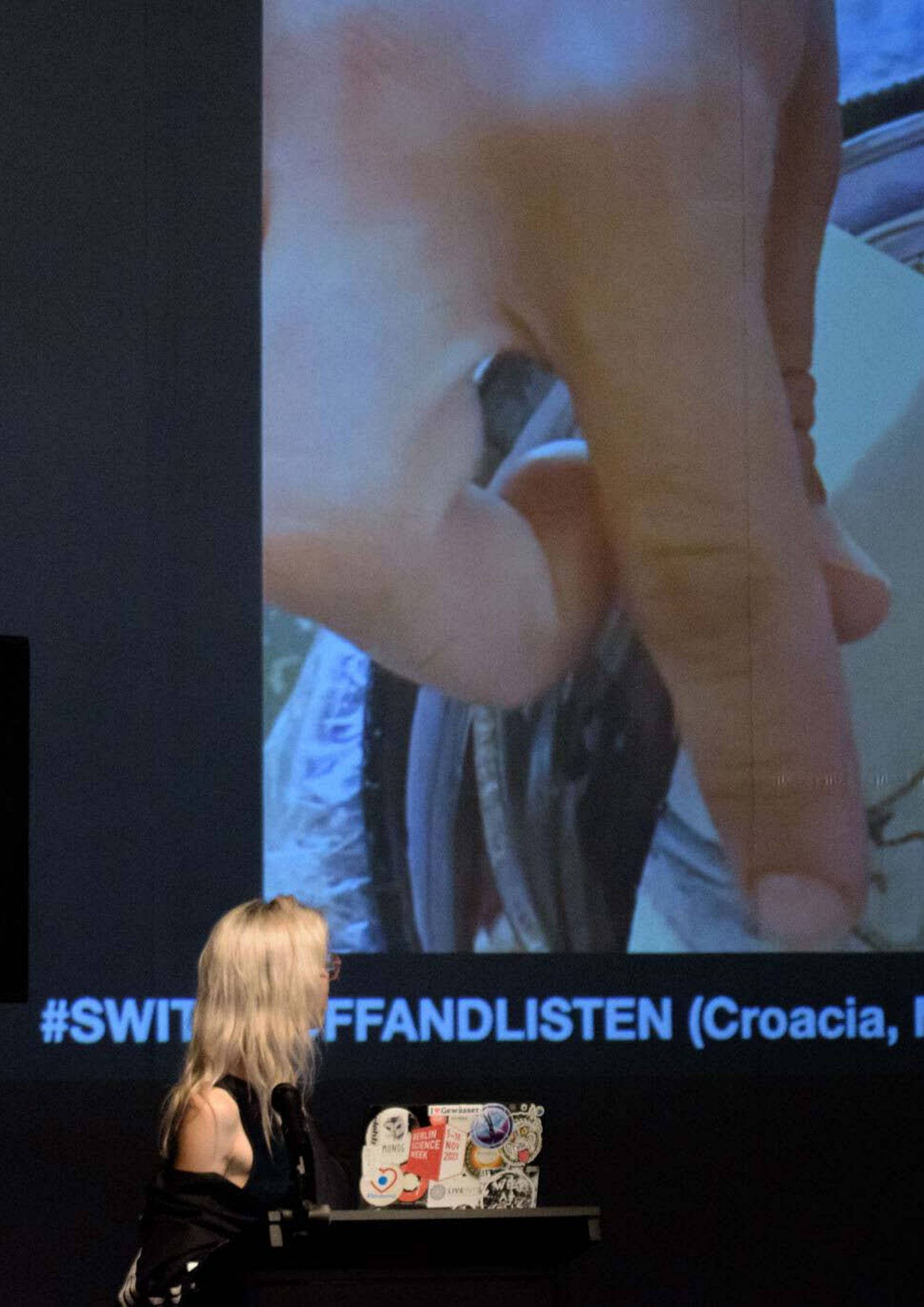
PHY

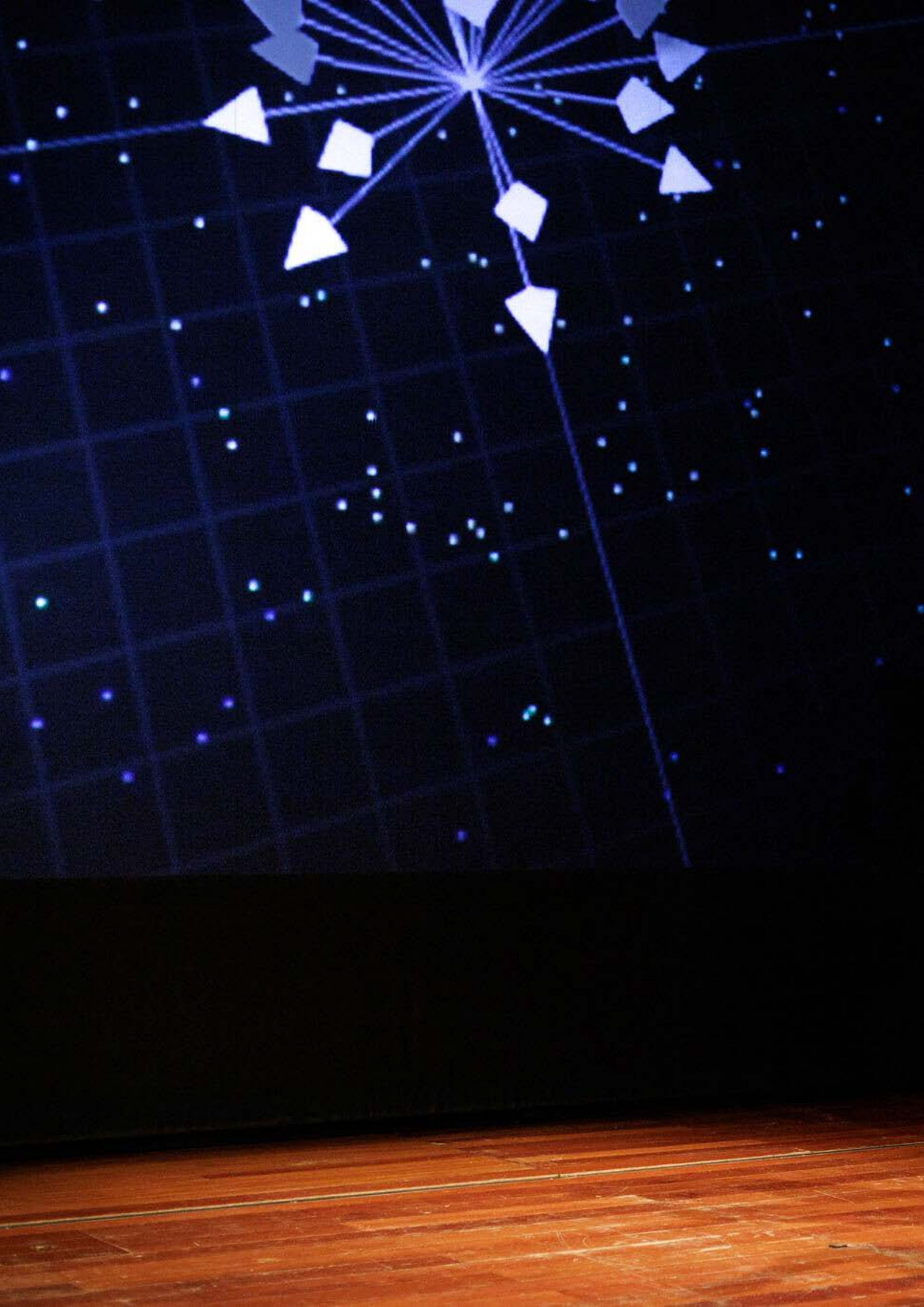
flickr


IMM
SOUND AND M
SMC 2024 / 4–6 Ju

P.PORTO

P.PORTO
P.PORTO


#SWIT FFANDLISTEN (Croacia,
BERLIN SCIENCE WEEK
Gewässer



P.PORTO

flickr

ACCESS TO ALL PHOTOS

This work was supported by FCT - Fundação para a Ciência e a Tecnologia, I.P., by project references UIDB/00693/2020 of CESEM – Centre for the Study of the Sociology and Aesthetics of Music of ESMAE-P.PORTO with DOI identifier 10.54499/UIDB/00693/2020 and LA/P/0132/2020 with DOI identifier 10.54499/LA/P/0132/2020

ISBN: 978-989-35207-5-8
DOI: 10.5281/zenodo.13918961

SMC 2024 / 4–6 July
ESMAE, Porto, Portugal

ESMAE ESCOLA SUPERIOR DE MÚSICA E ARTES DO ESPETÁCULO

CESEM

IN2PAST PATRIMÓNIO | ARTE | SUSTENTABILIDADE | TERRITÓRIO

fct Fundação para a Ciência e a Tecnologia

IMMERSIVE

SOUND AND MUSIC COMPUTING