

Musical Agents based on Self-Organizing Maps for Audio Applications

by

Kıvanç TATAR

M.Mus., İstanbul Technical University, 2014

B.Sc., Middle East Technical University, 2012

Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

in the
School of Interactive Arts and Technology
Faculty of Communication, Art and Technology

**© Kıvanç TATAR 2019
SIMON FRASER UNIVERSITY
Summer 2019**

Copyright in this work rests with the author. Please ensure that any reproduction or re-use
is done in accordance with the relevant national copyright legislation.

Approval

Name:	Kıvanç TATAR
Degree:	Doctor of Philosophy (Art and Technology)
Title:	Musical Agents based on Self-Organizing Maps for Audio Applications
Examining Committee:	Chair: Halil Erhan Associate Professor
	Philippe Pasquier Senior Supervisor Associate Professor
	Steve DiPaola Supervisor Professor
	Oliver Bown Supervisor Senior Lecturer Art & Design, University of New South Wales
	Thecla Schiphorst Internal Examiner Professor
	Matthew Yee-King External Examiner Lecturer Department of Computing, Goldsmiths, University of London
Date Defended:	May 28, 2019

Ethics Statement



The author, whose name appears on the title page of this work, has obtained, for the research described in this work, either:

- a. human research ethics approval from the Simon Fraser University Office of Research Ethics

or

- b. advance approval of the animal care protocol from the University Animal Care Committee of Simon Fraser University

or has conducted the research

- c. as a co-investigator, collaborator, or research assistant in a research project approved in advance.

A copy of the approval letter has been filed with the Theses Office of the University Library at the time of submission of this thesis or project.

The original application for approval and letter of approval are filed with the relevant offices. Inquiries may be directed to those authorities.

Simon Fraser University Library
Burnaby, British Columbia, Canada

Update Spring 2016

Abstract

Musical agents are artificial agents that tackle musical creative tasks. Musical agents implement the technologies of Artificial Intelligence (AI) and Multi-agent systems (MAS) for musical applications. Musical agent studies situate in the interdisciplinary studies of Musical Metacreation (MuMe) with a focus on the agent architectures. Metacreation and MuMe combine the artistic practice of Generative Arts with the scientific literature of Computational Creativity. We define Musical Metacreation as an interdisciplinary field that studies the partial or complete automation of musical tasks. In this work, we concentrate on an audio-based musical agent architecture with unsupervised learning while presenting the literature review of musical agents.

Our review of musical agents surveys seventy-eight musical agent systems that have been presented in peer-reviewed publications. Building on our literature review, we propose a typology of musical agents in nine dimensions of agent architectures, musical tasks, environment types, number of agents, number of agent roles, communication types, corpus types, input/output types, human interaction modality. Our typology of musical agents builds on the AI terminology and agent architecture typology in MAS. In comparison to agent typology of MAS, the categories that we present in our typology address the specific phenomenon that appear in the agent-based applications of musical tasks.

Our survey of musical agents indicated a possibility of research on an audio-based musical agent architecture with unsupervised learning. The implementations of musical agents that we present in this thesis utilize audio recordings with unsupervised learning because a variety of musical styles are available in the audio domain. Audio recordings are accessible; thus, the curation of a corpus for agent learning is easier. amount of work to gather the training data. To address this research possibility, we proposed an architecture called Musical Agent based on Self-Organizing Maps (MASOM) for audio applications. We were inspired by Edgard Varèse's definition of music, which suggested that music is "nothing but organized sounds." We put the notion of music as organized sounds into practice by combining autonomous audio latent space generation with musical structure modelling. This unique combination suggests that an audio-based musical agent architecture requires two kinds of sound organization: organizing sounds in latent sonic space to differentiate sound objects and organizing sounds in time to create temporal musical structures.

We present two main real-time applications of Musical Agents based on Self-Organizing Maps: architectures for experimental electronic music with machine listening and an architecture for virtual reality applications with respiratory user interaction. Our applications exemplify the strengths and possibilities of audio-based musical agents in the artistic domain. We believe that MASOM architectures can be useful for the applications of "musical creativity as it is." We also propose that the innovative perspective of MASOM architectures provide an exploration of the "musical creativity as it could be."

Keywords: Musical agents; Multi-Agent Systems; Artificial Intelligence; Musical Metacreation; Computational Creativity; Virtual Reality; Interactive Arts; Contemporary Arts

Dedication

I would like to dedicate this thesis to my family and friends. Notably, I want to thank my mother Gülsiraz Tatar and my father Yaşar Tatar for supporting me no matter how far my curiosity extends.

Acknowledgements

This research was funded by the Natural Sciences and Engineering Research Council of Canada Discovery programme, the Social Sciences and Humanities Research Council of Canada Insight programme.

Table of Contents

Approval	ii
Ethics Statement	iii
Abstract	iv
Dedication	vi
Acknowledgements	vii
Table of Contents	viii
List of Tables	xiii
List of Figures	xiv
1 Introduction and Overview	1
1.1 Introduction	2
1.2 Background	3
1.2.1 Generative Art and Computational Creativity	3
1.2.2 Creativity	4
1.2.3 Musical Style	5
1.2.4 Style Imitation	6
1.3 Motivations and Research Questions	7
1.4 Contributions	8
1.4.1 Publications	9
1.4.2 Artworks	13
1.5 Structure of Thesis	15
1.6 Summary	17
2 Musical Agents: a Typology and State of the Art towards Musical Metacreation	21
Abstract	21
2.1 Introduction	23

2.2	Generative Art and Computational Creativity	24
2.3	Typology of Musical Agents	26
2.4	Cognitive Musical Agents	33
2.4.1	Cognitive musical agents with knowledge representation	34
2.4.2	Cognitive musical agents with BDI architecture	35
2.4.3	Cognitive musical agents with cognitive models	37
2.5	Reactive Musical Agents	38
2.5.1	Reactive Musical Agents in Real-World Environments	39
2.5.2	Reactive Musical Agents in Virtual Environments	53
2.6	Hybrid Musical Agents	59
2.6.1	Hybrid musical agents using statistical sequence modelling	59
2.6.2	Hybrid musical agents combining statistical sequence modelling with rule-based models	66
2.6.3	Hybrid musical agents with Artificial Neural Networks	67
2.6.4	Hybrid Musical Agents with cognitive models	71
2.7	Evaluation of Musical Agents	75
2.7.1	Informal Evaluations	76
2.7.2	Formal Evaluations	76
2.7.3	Future Steps of Evaluation and Benchmarking	82
2.8	Ad Infinitum	84
2.8.1	Architectures and Algorithms	84
2.8.2	Interdisciplinarity of MuMe	85
2.8.3	Design Considerations	86
2.8.4	MuMefication	88
2.8.5	Challenges and Opportunities	89
2.9	Conclusion	91
3	MASOM: A Musical Agent Architecture based on Self-Organizing Maps, Affective Computing, and Variable Markov Models	115
Abstract		116
3.1	Introduction	117
3.2	Background	117
3.2.1	Self-Organizing Maps	117
3.2.2	Markov Models	119
3.2.3	Affective Computing	119
3.3	Related Work	120
3.3.1	Modelling Musical Memory with SOMs	120
3.3.2	Musical Agents with Markov Models	121
3.4	System Design	121

3.4.1	Sound Affect Estimation	121
3.4.2	The Learning in MASOM	125
3.4.3	The Generation in MASOM	129
3.5	Evaluation and Future Work	129
4	A Comparison of Statistical Sequence Models in Musical Agents based on Self-Organizing Maps	136
Abstract		137
4.1	Introduction	138
4.2	Related Work	139
4.3	System Architecture Details	139
4.4	Machine Listening	140
4.4.1	Segmentation	140
4.4.2	Thumbnailing	141
4.5	Organizing Sounds with Self-Organizing Maps	144
4.6	Musical Structure Generation with Statistical Sequence Modelling Algorithms . . .	146
4.6.1	Variable-Markov Models	146
4.6.2	Factor Oracle	149
4.6.3	Recurrent Neural Networks	149
4.7	Evaluation Methodology	150
4.7.1	N-gram Cloning	150
4.7.2	Repetition Percentage Analysis	151
4.7.3	The Longest Sub-sequence Cloning	151
4.8	Results	151
4.8.1	Corpora	151
4.8.2	Generated Sequences	154
4.8.3	Analyses	154
4.9	Discussions	155
4.9.1	How much cloning is acceptable?	155
4.9.2	How much repetition is preferable?	159
4.9.3	Sparsity	159
4.9.4	Curse of Dimensionality	159
4.10	Conclusion	160
5	Audio-based Musical Artificial Intelligence and Audio-Reactive Visual Agents in Revive	166
Abstract		167
5.1	Introduction	168

5.2	Multi-agent Systems in Creative Artificial Intelligence for Music and Multimedia	169
5.3	The Performance Setup	170
5.4	Audio-reactive visual agents in Revive	170
5.5	Sonic Strategies in Revive	173
5.5.1	Musical Artificial Intelligence, MASOM-FO	174
5.5.2	Distruption of fixed-media	179
5.5.3	3D audio spatialization techniques in Revive	181
5.6	Conclusion	182
6	Respire: A Virtual Reality Art Piece with a Musical Agent guided by Respiratory Interaction	185
Abstract		186
6.1	Immersive Environment of Respire	187
6.2	Body and breath in immersive virtual environments	187
6.3	Artificial Intelligence, Multi-Agent Systems, and Musical Agents	188
6.4	Affect Recognition	189
6.5	System Details	189
6.5.1	Sound Memory	190
6.5.2	Signal Processing of Breathing Data	191
6.5.3	Generative Algorithm of Musical Agent	192
6.6	Exhibitions	193
6.7	Next Steps	195
7	Summary and Conclusion	198
7.1	Summary	199
7.2	Limitations	200
7.3	Future Work	204
7.3.1	Automatic Latent Space Generation for Audio	204
7.3.2	Hierarchical Models for Audio-based Musical Agents	205
7.3.3	Automatic Spatialization of Musical Agents	206
7.3.4	Visualization and Embodiment Musical Agents	206
7.4	Final Word	207
Cumulative Bibliography		210
Appendix A	Revive: An Audio-Visual Performance with Musical and Visual Artificial Intelligence Agents	232
Appendix B	Respire: A Breath Away from the Experience in Virtual Environment	239

Appendix C Automatic Synthesizer Preset Generation with PresetGen	246
Appendix D Ranking Based Experimental Music Emotion Recognition	278
Appendix E Quantitative Analysis of the Impact of Mixing on Perceived Emotion of Soundscape Recordings	287
Appendix F Attending to Breath: Exploring How the Cues in a Virtual Environment Guide the Attention to Breath and Shape the Quality of Experience to Support Mindfulness	294
Appendix G The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality	309

List of Tables

Table 1.1	Main Contributions	10
Table 1.2	The contributions of the side projects	11
Table 2.1	Musical Agents	28

List of Figures

Figure 2.1	The continuum of autonomy	25
Figure 2.2	The 9 dimensions of our musical agents typology	27
Figure 2.3	The block diagram of the system ③ in Table 2.1 [24]	34
Figure 2.4	An example of role assessment in MUSIC-MAS [145]	36
Figure 2.5	The architecture of MusiCOG [128]	37
Figure 2.6	Bob’s system architecture [182]	40
Figure 2.7	The architecture of LL [50]	42
Figure 2.8	The structure of BBCut2 [48]	44
Figure 2.9	<i>ParamBOT</i> , a curator agent implemented using the Musebot framework in MAX [72]	47
Figure 2.10	The block diagram of GenJam [22]	49
Figure 2.11	The system architecture of Yee-King’s [204] musical agent	50
Figure 2.12	The block diagram of Kinectic Engine version 3 [68]	51
Figure 2.13	The system design of <i>Petri</i> [20]	54
Figure 2.14	Energy-based generation in Ringomatic [12]	60
Figure 2.15	The block diagram of <i>OMAX</i> [115]	61
Figure 2.16	The improvisation renderer in Improtex [148]	63
Figure 2.17	The interface of PyOracle	65
Figure 2.18	The Subsumption architecture of the Reactive Accompanist [42]	68
Figure 2.19	The architecture of Mocking-bird [118]	72
Figure 2.20	The system architecture of <i>MASOM</i> [177]	74
Figure 2.21	The number of musical agents per architecture type	82
Figure 2.22	The continuum of autonomy in musical agent design	86
Figure 3.1	The generation interface of <i>MASOM</i> includes a visualization of SOM. The visualization shows one dimension at a time. The dimensions are normalized between -1.0 (black) and 1.0 (white) for the visualization.	117
Figure 3.2	Each dot represents a novel segment of Stockhausen’s <i>Kontakte</i> . Each segment is labeled using the dimensional affect estimation model of <i>MASOM</i>	122
Figure 3.3	The architecture of <i>MASOM</i>	123

Figure 3.4	The waveform is the seventh track of Bernard Parmegiani's <i>De Natura Sonorum</i> album. Each square indicates a novel segment. The corresponding SOM node of the segment is written inside the box.	125
Figure 4.1	The user interface of MASOM's learning algorithms	139
Figure 4.2	The training of MASOM: a) Segmentation b) Labelling the audio samples c) Sound memory where squares stand for SOM nodes that are a clusters of audio samples d) Creating a symbolic representation of the original song using the clusters indexes of audio samples e) statistical sequence model to learn temporal transitions.	146
Figure 4.3	Factor Oracle generated using the sequence <i>ABAAB</i>	148
Figure 4.4	The dimensions of SOM trained on the audio segments of Electroacoustic music corpus. Each figure shows one audio feature listed in Section 4.4.2. The black and white represent the minimum and the maximum values, respectively.	151
Figure 4.5	The dimensions of SOM trained on the audio segments of repetitive experimental electronic music corpus. Each figure shows one audio feature listed in Section 4.4.2. The black and white represent the minimum and the maximum values, respectively.	152
Figure 4.6	The percentage of n-gram cloning in the generated sequences using the corpora of electroacoustic music.	155
Figure 4.7	The percentage of n-gram cloning in the generated sequences using the corpora of repetitive experimental music.	155
Figure 4.8	The percentage of repetition for Electroacoustic music corpus	156
Figure 4.9	The percentage of repetition for IDM music corpus	156
Figure 4.10	The box-plot of the longest subsequence cloning in the generated sequences are calculated using the corpora of electroacoustic music. The red line represent the mean, while the circles indicate the outliers.	157
Figure 4.11	These statistics of the longest subsequence cloning in the generated sequences are computed using the corpora of IDM music. The red lines represent the mean, while the circles indicate the outliers.	157
Figure 5.1	The UI of Revive's performance setup	170
Figure 5.2	Three snapshots of dome views illustrate three types of particle engine renderings. In each figure, three distinct colors corresponds to three visual agents that react to three sonic performers.	171

Figure 5.3	The offline learning in MASOM: a) Segmentation b) Labelling the audio samples c) Sound memory where squares stand for SOM nodes that are a clusters of audio samples d) Creating a symbolic representation of the original song using the clusters indexes of audio samples e) statistical sequence model to learn temporal transitions.	174
Figure 5.4	Factor Oracle generated using the sequence <i>ABAAB</i>	177
Figure 5.5	The framework for using fixed-media compositions as content for the live audio-visuals with 3D audio: a) the game controller interface and the mappings b) wavetable synthesis c) 3D audio visualization with the SAT dome setup.	178
Figure 6.1	The virtual environment visuals in <i>Respire</i>	186
Figure 6.2	The system architecture of <i>Respire</i>	188
Figure 6.3	The system architecture of <i>Respire</i>	189
Figure 6.4	The musical agent in <i>Respire</i>	190
Figure 6.5	A dancer in still position in P.O.E.M.A. (Photo: Adriano Fagundes.) . . .	192
Figure 6.6	A user with VR headset, a dancer, and spectators in the exhibition space of P.O.E.M.A. (Photo: Adriano Fagundes.)	193
Figure 6.7	A dancer in movement in P.O.E.M.A. (Photo: Adriano Fagundes.) . . .	194
Figure 7.1	Revive performance at the Société des Arts Technologique during the festival MUTEK Montreal 2018, photo credit: Ashley Gesner	200
Figure 7.2	Revive members from left to right, Remy Siu, Kivanç Tatar, and Philippe Pasquier, photo credit: Ashley Gesner	201
Figure 7.3	The audio-reactive particle engine in Revive's visuals, photo credit: Ashley Gesner	202

Chapter 1

Introduction and Overview

1.1 Introduction

Music technology influences how we create music. Many examples where new tools for musical tasks have led to new aesthetics and styles, are noticeable in history. The invention of electric and electronics resulted in the technologies of audio recording, reproduction, and synthesis. Early electronic music composers explored the electronic medium while building on the conventional music theory. Along with the discovery of the electronic medium, the Futurists such as Luigi Russolo [13] expanded the sound palette of music to all sounds [6]. Early electronic music composers pushed their ideas towards a new generalized understanding of music, that is “nothing but organized sounds” [27, 22, 21, 18, 19].

The history of early electronic music exemplifies how new technology creates new artistic possibilities. For example, Pierre Boulez carefully edited vocal recordings on tape to create melodic jumps that are impossible to be performed within the human vocal capabilities. In Boulez’s compositions, these vocal recordings sounded as if they were performed in a recording session [6]. Another example where music technology created new artistic possibilities is the mobile recording technologies that allowed electroacoustic music composers to record sounds outside of the studio. Hence, the mobility advancements in recording technologies resulted in the soundscape compositions in electroacoustic music [26]. These two examples demonstrate the advancements in technology resonate in the artistic practices. Likewise, the rise in Artificial Intelligence (AI) and Multi-agent Systems (MAS) have created new opportunities to expand our understanding of musical creativity.

Musical Agents are agent-based systems that implement the technologies of Machine Learning, Artificial Intelligence (AI), and Multi-Agent Systems (MAS) for musical applications. The notion of agents appears in many disciplines such as Computer Science, Social Sciences, Philosophy, and Cognitive Science. In Computer Science, Wooldridge [30] defines an agent as an autonomous system that initiates actions to respond to its environment in timely fashion. In Chapter 2, we revisited this definition while taking the musical tasks into account. Hence, we define musical agents as autonomous software agents that tackle musical creative tasks. The musical agent systems research MAS technologies to solve problems of musical creativity, and discover new questions. Musical agents systems that we survey in Chapter 2 exhibit agent behaviours of autonomy, reactivity, proactivity, interactivity, adaptability, coordination, communication, and emergence.

In addition to the terminology and typology of musical agents in Chapter 2, this thesis proposes an audio-based musical agent architecture with unsupervised learning, called Musical Agent based on Self-Organizing Maps (MASOM) in Chapter 3 and 4. MASOM’s architecture differs from previous audio-based musical agent implementations such as OMAX [2, 12], Improtex [15], Audio Oracle [10], Variable Markov Oracle [28], FILTER [16], and Mocking-Bird [14] by increasing the training data size from the scale of several minutes to several hours. The capability of increasing the training data to mid-sized and big-sized data aims to contribute to the musical agent literature by moving towards musical agent architecture that could listen to more music than a human could [7].

MASOM’s musical applications span machine improvisation, composition, and computer-assisted composition. Machine improvisation integrates musical improvisation within a musical agent architecture. Musical agents applying machine improvisation unify interactive behaviours with machine listening. These systems can perform live on stage while interacting with other human or non-human musicians. The machine improvisation systems that we survey in Chapter 2 mainly apply free improvisation that is non-idiomatic improvised music. In addition to machine improvisation applications, MASOM can generate musical output on the fly without any input. This naturally creates the possibilities of musical applications where MASOM assists composition process by generating audio for musical composition.

Three types of artificial agents exist in MAS: software agents that are purely computational, virtual agents that are embodied in a Computer Generated Image (CGI), and robotic agents. In this thesis, Chapter 2, 3, 4 focus on software agents. Chapter 5 delves into visualization of musical agents using audio-reactive generative visuals. Chapter 6 embeds a musical agent within a Virtual Reality environment. We exclude the musical agent applications of robotics, and Chapter 7 concludes this thesis while mentioning the embodiment of musical agents with robotics as future work.

In the context of musical agents, our understanding of music is inclusive of any art discipline that incorporates sound medium. Furthermore, our approach to music is in line with the electroacoustic and experimental electronic music theories. These theories propose that music is “nothing but organized sound” [27] involving multiple layers [22]. Any sound can be used to produce music [13, 27], and strong connections exist between pitch, noise, timbre, and rhythm [22, 21, 18, 19]. The timescales of music span nine levels from the shortest to the longest: infinitesimal, subsample, sample, sound object, meso, macro, supra, and infinite [18]. Sounds physically exist in a 3D space and sound motion is another dimension of musical composition and performance [22]. The artworks that we list in Section 1.4.2 span experimental electronic music styles such as noise music, glitch, electroacoustic music, Intelligent Dance Music (IDM), mainstream electronic music, and ambient music. We believe that the theories of experimental electronic music provide a generalized framework to formulate, analyze, and model all musical styles for musical agent applications.

In the following, we situate musical agent research at the intersection of Generative Art and Computational Creativity.

1.2 Background

1.2.1 Generative Art and Computational Creativity

The strong connection between art and technology is in the intersection of Science and creativity studies. Musical agents is an interdisciplinary field where the scientific literature of Computational Creativity is interconnected with the creative applications of Generative Arts. Galanter [11] defines Generative Art as,

Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural

invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.

In this definition, “Art” refers to all artistic practices including music. Musical Metacreation (MuMe) applies the scientific terminology of Computational Creativity literature on creativity to the autonomous systems making music. Colton and Wiggins [8] define Computational Creativity as “...The philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative.”

The notion of Metacreation resonates back to 1950s, as Nicolas Schöffer mentions [29]:

We are no longer creating a work, we are creating creation. ...We are able to bring forth...results...which go beyond the intentions of their originators, and this in infinite number.

Buchanan [5] also mentioned the notion of “creativity at the meta-level.” Explicitly, the term Metacreation appeared in Mitchel Whitelaw’s [29] MIT Press book titled *Metacreation: art and artificial life*.

Metacreation is an interdisciplinary field that studies and produces systems that partially or completely automate creative tasks [24]. As such, Metacreation encompasses Generative Arts (an artistic practice) and Computational Creativity (a scientific research practice). Hence, Metacreation or any terminology that we propose to define this concept englobes both artistic and scientific works. Building on the literature of Metacreation, Pasquier et al. [17] define Musical Metacreation as “...a subfield of computational creativity that addresses music-related creative tasks.” We revisit this definition in Chapter 2, and we propose that MuMe is the partial or complete automation of musical tasks [24]. MuMe is an interdisciplinary field that incorporates both artistic and scientific terminology for autonomous systems making music. Musical Metacreation encompasses machine musicianship, algorithmic music, generative music, machine improvisation, or other appellations that refer to the idea of MuMe. Chapter 2 proposed MuMe as a field that aims to bring together all fields that apply autonomous approaches for creative musical tasks while applying the scientific terminology of Computational Creativity to the autonomous systems making music. We think that the literature of Computational Creativity gives an established ground to explain whether MuMe systems are musically creative and if they are, the kind of creativity that MuMe systems output.

1.2.2 Creativity

Boden [3] defines creativity as “the ability to generate new forms.” This definition explains creativity by focusing on the artifact. Boden continues by proposing *psychological* and *biological* creativity to categorize human and non-human creativity. *Biological* creativity is “the ability to generate new cells, organs, organisms, or species.” In comparison, *psychological* creativity is “the ability to generate ideas and/or artifacts that are new, surprising, and valuable.”

Boden focuses on two key points to understand which forms are new. First, Boden discusses the notion of *novelty* in creativity. Second, *historical creativity* is a special case of psychological

creativity in which generated form is novel to the community. Furthermore, Boden states three types of creativity as a result of *mechanisms* generating novelty; *exploratory*, *combinational*, and *transformational*. First, *exploratory* creativity is making novel forms that satisfy constraints of a particular style. An example of exploratory creativity is improvising a Jazz melody in the style of Charlie Parker. Second, *combinational* creativity is combining styles in novel ways such as improvising a Jazz melody in the style of Chet Baker with the ornamentations of Charlie Parker. Third, *transformational* creativity is the expansion of known conceptual space. An example of transformational creativity is John Cage's idea of using random sounds of the audience as material for musical performance.

The notion of conceptual space in creativity helps the comprehension of style. In the context of this thesis, musical agents that employ unsupervised learning (such as MASOM) are naturally connected to the concept of musical style. In the following, we continue with explaining style and musical style.

1.2.3 Musical Style

Defining style is not an easy task. The term style appears in variety of disciplines such as the style of writing, or an artist, an artwork, a genre, or an artisan. Oxford Dictionary (2017) defines style as “a particular procedure by which something is done; a manner or way.” Although this definition also applies to music, we require more specific description musical style to understand computational systems that addresses problems and tasks of musical style. Dannenberg [9] clarifies six usage of the term style in music:

1. The style of a historical music era
2. The style of a composer
3. The style of a performer
4. The style of a musical texture
5. The style of music emotion associations
6. The style of a genre

In the context of seventy-eight musical agents that we survey in Chapter 2, musical style is a combination of the second, third, fourth, and the fifth usages of the term style in Dannenberg's [9] list. In the case of MASOM, the agent learns the style of a composer or performer using statistical sequence models (see Chapter 5). The agent captures the style of the musical texture by generating a latent audio space using the audio segments in the corpus (see Section 4.5). MASOM also learns music emotion associations by incorporating a Music Emotion Recognition (MER) algorithm in the agent's machine listening module (see Section 3.4.1). Purely generative music systems mainly focus on the first and the last usages of style in the [9]'s list by applying exhaustive search techniques

that are not necessarily running online [4]. We elaborate on the differentiation of purely generative systems and musical agents in Section 2.1.

While Dannenberg [9] focuses on practical usage of the term, Siefkes [20] describes style using a semiotic model as “a type of sign process with certain properties.” Siefkes continues,

The [semiotic] model consists of two main parts, corresponding to two sign processes that interact when a style is produced or when it is received: (1) Style is created when choice on the basis of a schema takes place and when regularities in this choice appear. These regularities can be formulated as feature rules. The first sign process describes the inscription of these rules (by a style producer) into a realisation of a schema, and the readout (by a style receiver) out of the realisation. (2) On the basis of the stylistic features used in the first sign process, a stylistic interpretation can be created by the style receiver and also envisaged and taken into account by the style producer. Both processes interact in creating stylistic information.

Hence, Siefkes indicates two types of style: perceived style and expressed style; while focusing on two subjects that appear in the creative process: the creator and the recipient. For example, the composer’s understanding of a particular style does not necessarily the same as the recipient’s one. The comprehension of style in musical agent literature in Chapter 2 does not necessarily differentiate the perceived style from the expressed style.

Style also refers to a set of predefined attributions of form. These attributions are shared within a particular style. For example, Baroque Music is contrapuntal, ornamented, with frequent, less meaningful modulation, has single vivid feeling, and keeps constant intensity throughout whereas Classical Music is homophonic, emphasizes formal structure, uses modulation as a structural element, explores a range of emotions within a piece, and applies a dramatic climax and a resolution [9]. If we formalize this definition of style, we can define style in the conceptual space in which all possible forms exist.

1.2.4 Style Imitation

Style imitation is one of nine musical tasks that musical agents implement (see Chapter 2). Audio-based musical agents with unsupervised learning are naturally connected to the task musical style imitation. The training set of musical agents delineates a particular style and conceptual space to the agent. We can define a style as a region in the conceptual space and explore new forms within this region, which corresponds to exploratory creativity. This is also referred as *style imitation*. Pasquier et al. [17] formalizes style imitation,

Given a corpus $C = C_1, \dots, C_n$ representative of style S , style imitation is to generate new instances that would be classified as belonging to S by an unbiased observer (typically a set of human subjects).

Although style imitation aims to apply “creativity as it is” [17], it has been shown that the algorithms that we use for style imitation introduce biases [25]. Hence, the work in thesis embraces these biases, proposing that the bias introduced by an algorithm can power the musical aesthetics (see Chapter 4). We hope that this thesis initiates a discussion of how the bias of the algorithm can be a source for exploring “creativity as it could be.”

1.3 Motivations and Research Questions

The technologies of Multi-Agent Systems (MAS) and Artificial Intelligence (AI) are beneficial for style imitation and creative applications in general because MAS are distributed/concurrent systems that are autonomous, able to make independent decisions, and run online [30]. Likewise, musical performance and improvisation involve distributed, concurrent musicians that communicate with each other. Musical performance emerges from the coordination and communication of musicians. In musical composition, the creative process utilizes distributed and concurrent compositional layers that are in a complex relationships with each other [19].

The applications of MAS are beneficial to modelling and designing musical creativity because musical creativity involves distributed, coordinated entities with perception and action abilities. Software agents in MAS also have perception and action abilities. Likewise, in a live music performance, musicians collaboratively create music by listening to each other. Distributed, modular, and autonomous architectures of Multi-agent Systems provide promising opportunities to model these musical tasks.

Audio-based musical agents with unsupervised learning provide flexible architectures in which the musical style of agent’s output depends on the training corpus. The curation of training corpus has a direct effect on the agent, and this flexibility allows artists to utilize same architecture for various applications. For example, the artwork created for this thesis applied the MASOM architecture (see Section 3) in the context of noise music, glitch, electroacoustic music, Intelligent Dance Music (IDM), mainstream electronic music, ambient music within the artworks of live musical performances, virtual reality applications, gallery installations, and live audio-visual performances (see Section 1.4.2).

Our research focuses on audio as opposed to symbolic representations of music such as Musical Instrument Digital Interface (MIDI) or musical scores. Audio recordings are the final stages of any musical production. The recordings inherit all musical information that is put into the production. Symbolic representations may exclude musical information such as performers’ interpretations or timbral transitions. Given that timbral transitions or spectromorphology is crucial for electroacoustic music, this thesis focus on audio-based music agents for audio applications.

The flexibility provided by the unsupervised learning helps artists to create content in a fast manner. For example, we trained MASOM on the previous compositions Mehmet Ünal for our performance at the Ars Electronica Festival 2017. Then, we used the trained agent to generate audio recordings in the scale of several hours. The composer in the collaboration, Mehmet Ünal used the

generated recordings to compose a 40-min audio-visual composition. The utilization of MASOM in the collaboration allowed the composer to start from draft compositions instead of starting from scratch.

In addition to the content creation advantages, audio-based musical agents also elicit new artistic opportunities. For example, the interactivity in MASOM utilizes a machine listening algorithm with affective computing, and this allowed the artist to improvise freely with a musical agent that is trained on the artist's own style. To test this idea, we initially trained the MASOM on Kivanç Tatar's noise album during our early experiments¹. Later, we incorporated MASOM architecture into an audiovisual artwork in which we turned fixed media compositions of well-known electroacoustic music composers into musical agents that perform live on stage (please refer to Chapter 5). Towards "creativity as it could be" [17, 24], these artistic applications are some examples of many opportunities created by audio-based musical agent architectures with unsupervised learning.

Building on these motivations, this thesis focus on the following four research questions,

- **RQ1:** How to characterise musical agents?
- **RQ2:** How to design a musical agent architecture that learns musical forms by listening to audio recordings of music?
- **RQ3:** How to integrate a real-time machine listening algorithm into an audio-based musical agent architecture for live performances?
- **RQ4:** How to incorporate the audio-based musical agent architecture using unsupervised learning into artistic applications?

1.4 Contributions

In relation to the first research question (**RQ1**), we surveyed 78 musical agent systems that are presented in peer-reviewed publications. Our survey contributes to the field by proposing a typology of musical agents including nine dimensions of agent architectures, musical tasks, environment types, number of agents, number of agent roles, communication types, corpus types, input/output (I/O) types, human interaction modality (HIM). This typology is built upon the literature of Multi-agent systems and agent architectures in Computer Science, while taking the specificities of musical tasks and applications into consideration. In addition to the typology of musical agents, we also propose a typology of evaluation methodologies for musical agent research.

Our survey of musical agents suggested a research direction of audio-based musical agents using unsupervised learning using an audio recording data that is in the scale of hours of music. To answer the second research question (**RQ2**), we proposed a novel musical agent architecture, MASOM that combines an unsupervised machine learning algorithm to organize the audio memory

¹<https://kivanctatar.com/MASOM-0-01>

with statistical sequence models to capture the musical temporal structure. We aimed for a flexible algorithm that would perform successfully for any topology given that the topology of audio corpus in the feature space varies greatly depending on the selected music recordings. The proposed musical agent architecture can be trained on a dataset that is in the scale of hours of music. In regards to the third research question (**RQ3**), the machine listening of MASOM incorporates high level musical features such as affective features. The flexibility of performing alone or with multiple software and/or human agents also plays an important role for various multimedia applications. In the machine learning for music domain, some researchers pointed out the lack of deployment of autonomous music systems to real-world applications [23]. Regarding the fourth research question (**RQ4**), we incorporated the musical agent architectures proposed in this thesis into live musical performances, an audio-visual live performance project, and a virtual reality artwork. The details and dates of public presentations of these artworks are available in Section 1.4.2

In the following, we specify all contributions of this doctoral study and clarify the thesis structure. Table 1.1 and 1.2 points out contributions of the thesis chapters and their relation to the publications and artworks listed below. Table 1.1 also clarifies the connection of contributions in thesis chapters to the research questions listed in Section 1.3.

1.4.1 Publications

The contributions of this cumulative thesis emerged as peer-reviewed journal papers and conference papers listed below, as well as art pieces, and public performances listed in Section 1.4.2. **P#** indicates the index of the main publications while **PA#** marks the publications related to the works listed in the appendixes. Table 1.1 and 1.2 clarify the contributions of publications listed below, and their connection to the research questions and the artworks.

1. **P1:** Kivanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 47(4), 1–50, 2018. URL doi.org/10.1080/09298215.2018.1511736
2. **P2:** Kivanç Tatar and Philippe Pasquier. MASOM: A Musical Agent Architecture based on Self-Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation*, 2017.
3. **P3:** Kivanç Tatar, Jeff Ens, Jonas Krasch, Jianyu Fan and Philippe Pasquier. A Comparison of Statistical Sequence Models in Musical Agents based on Self-Organizing Maps. To be submitted.
4. **P4:** Kivanç Tatar & Philippe Pasquier, and Remy Siu. Audio-based Musical Artificial Intelligence and Audio-Reactive Visual Agents in Revive. Accepted to the *International Computer Music Conference and New York City Electroacoustic Music Festival 2019* (ICMC-NYCEMF 2019).

Chapter Contributions

Publication	Artworks
<ul style="list-style-type: none"> - Introduction to Generative Arts and Computational Creativity - Motivations and Research Questions - Contributions 	—
<ul style="list-style-type: none"> - Survey of 78 musical agent systems - A typology of musical agents (RQ1) - A typology of evaluation methodologies - Future directions in Musical Metacreation 	<p>P1 - Journal of New Music Research</p>
<ul style="list-style-type: none"> - An audio-based musical agent system with unsupervised learning and machine listening (RQ3) - Combining automatic audio latent space generation with a statistical sequence model (RQ2) 	<p>P2 - International Workshop on Musical Metacreation</p>
<ul style="list-style-type: none"> - A comparison of four Machine Learning models for musical structure modelling in MASOM (RQ2) - An analytical study with three measures of n-gram cloning, repetition percentage analysis, and the longest sub-sequence cloning (RQ2) 	<p>P3 -</p>
<ul style="list-style-type: none"> - Incorporating a musical agent into a live audio-visual performance (RQ4) - A framework with automatized cues to carry out structured improvisation (RQ4) 	<p>P4 - International Computer Music Conference</p>
<ul style="list-style-type: none"> - An immersive art piece that brings together three components: <ul style="list-style-type: none"> - An immersive virtual reality environment, - Embodied interaction (via a breathing sensor), - Musical agent system to generate unique experiences of augmented breathing (RQ4) 	<p>P5 - Leonardo Music Journal</p>

Table 1.1: Main Contributions

Appendix

Contributions in Relation to the Thesis

Publication	Artworks
A - Revive: An Audio-Visual Performance with Musical and Visual Artificial Intelligence Agents	<ul style="list-style-type: none"> - Initial specifications of a live audio-visual performance incorporating a musical agent - Chapter 5 builds on these specifications
B - Respire: A Breath Away from the Experience in Virtual Environment	<ul style="list-style-type: none"> - Initial description of a virtual environment that inherits a musical agent that reacts to user's breathing - Chapter 6 builds on this paper by fully disclosing the system details
C - Automatic Synthesizer Preset Generation with PresetGen	<ul style="list-style-type: none"> - Understanding sound similarity and related audio features to calculate sound similarity - In relation to the sound similarity in Section 4.4 and 4.5 - Synthesizing sounds using audio feature vectors - Related to the Chapter 7 and Future Work
D - Ranking Based Experimental Music Emotion Recognition	<ul style="list-style-type: none"> - Music Emotion Recognition for Experimental Music using Rank-Support Vector Machines - In connection with the Chapter 7 and Future Work
E - Quantitative Analysis of the Impact of Mixing on Perceived Emotion of Soundscape Recordings	<ul style="list-style-type: none"> - Computation of the perceived emotion of the mixed-soundscape recordings based on the perceived emotion of source soundscape recordings - The details of the Music Emotion Recognition model that is incorporated in MASOM (Section 3.4.1, 4.4, and 6.4)
F - Attending to Breath: Exploring How the Cues in a Virtual Environment Guide the Attention to Breath and Shape the Quality of Experience to Support Mindfulness	<ul style="list-style-type: none"> - An immersive virtual environment with a generative soundtrack that supports sustained attention on breathing by employing the users' breathing in interaction - Micro-phenomenology interviews to unfold the process in which breath awareness can be induced and sustained in this environment - Respire system described in Chapter 6 is used as a research tool in this work
G - The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality	<ul style="list-style-type: none"> - How two different mappings (metaphoric, and "reverse") of embodied interaction design might enhance the affective properties of an immersive virtual environment using a musical agent - Respire system in Chapter 6 builds on the Pulse.Breath.Water
PA1 - CHI Conference on Human Factors in Computing Systems	PA1 - CHI Conference on Human Factors in Computing Systems
PA2 - CHI Conference on Human Factors in Computing Systems	PA2 - CHI Conference on Human Factors in Computing Systems
PA3 - Journal of New Music Research	PA3 - Journal of New Music Research
PA4 - International Symposium for Music Information Retrieval	PA4 - International Symposium for Music Information Retrieval
PA5 - Sound and Music Computing Conference	PA5 - Sound and Music Computing Conference
PA6 - Designing Interactive Systems Conference	PA6 - Designing Interactive Systems Conference
PA7 - International Conference on Human-Computer Interaction	PA7 - International Conference on Human-Computer Interaction

Table 1.2: The contributions of the side projects

5. **P5:** Kivanç Tatar, Mirjana Prpa, and Philippe Pasquier. *Respire: A Virtual Reality Art Piece with a Musical Agent guided by Respiratory Interaction*. *Leonardo Music Journal*, in press.
6. **PA1:** Kivanç Tatar, Philippe Pasquier, and Remy Siu. (2018). REVIVE: An Audio-Visual Performance with Musical and Visual Artificial Intelligence Agents. In *CHI EA '18 Extended Abstracts (ArtCHI) of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1–6). Montreal, Canada: ACM Press. <https://doi.org/10.1145/3170427.3177771>
7. **PA2:** Prpa M., Tatar K., Schiphorst T., and Pasquier P. (2018). Respire: A Breath Away from the Experience in Virtual Environment. In *CHI EA '18 Extended Abstracts (ArtCHI) of the 2018 CHI Conference on Human Factors in Computing Systems* (pp. 1-6). Montreal, Canada: ACM Press. <https://doi.org/10.1145/3170427.3180282>
8. **PA3:** Kivanç Tatar, Mathieu Macret, and Philippe Pasquier. Automatic Synthesizer Preset Generation with PresetGen. *Journal of New Music Research*, 45(2), 124-144, 2016. doi. <https://doi.org/10.1080/09298215.2016.1175481>
9. **PA4:** Jianyu Fan, Kivanç Tatar, Miles Thorogood, and Philippe Pasquier. Ranking Based Experimental Music Emotion Recognition. In *Proceedings of the 18th International Society for Music Information Retrieval Conference*, ISMIR 2017.
10. **PA5:** Jianyu Fan, Miles Thorogood, Kivanç Tatar, Philippe Pasquier. Quantitative Analysis of the Impact of Mixing on Perceived Emotion of Soundscape Recordings. In *Proceedings of Sound and Music Computing*, SMC 2018.
11. **PA6:** Mirjana Prpa, Kivanç Tatar, Jules Françoise, Bernard Riecke, Thecla Schiphorts, and Philippe Pasquier. Attending to Breath: Exploring How the Cues in a Virtual Environment Guide the Attention to Breath and Shape the Quality of Experience to Support Mindfulness. In *Proceedings of the 2018 Designing Interactive Systems Conference* (pp. 71-84). ACM Press. 2018 <https://doi.org/10.1145/3196709.3196765>
12. **PA7:** Mirjana Prpa, Kivanç Tatar, Bernard Riecke, and Philippe Pasquier. (2017). The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality. In the 19th International Conference on Human-Computer Interaction, Vancouver, BC, Canada.

1.4.2 Artworks

We produced artworks and performances using the knowledge created in the above mentioned publications. In the following, we list these artworks and performances that are presented to the public. Each item starts with the date of the public presentation, and continues with the artwork description that mentions the collaborators, venues or festivals, and where the artwork was presented to the public. The items end with Kivanç Tatar's role in the artwork and the links to the documentations. Please refer to the tables 1.1 and 1.2 for the connection of these artworks to the research questions, contributions, and publications.

- **A1:** 09/02/2019 – *REVIVE* Concert at the Living Things 2019 with Philippe Pasquier, at Kelowna Art Gallery (6 speakers), Kelowna, BC, Canada – Role: Performer & Developer – Link: <https://kivanctatar.com/revive-living-things-2019>
- **A2:** 12/2018 – *ZETA* touch-based interactive installation for 360 immersive interface with multichannel audio, at the Immersive Lab at the Institute for Computer Music and Sound Technologies, Zürich University of the Arts with Philippe Pasquier; Zürich, Switzerland – Role: Artist & Developer – Link: <https://kivanctatar.com/zeta>
- **A3:** 08/12/2018 – *REVIVE* Concert at Zurich University of the Arts with Philippe Pasquier and Remy Siu at the Konzertsaal 1 (26 speakers and 4 subwoofers), Zürich, Switzerland – Role: Performer & Developer – Link: <https://kivanctatar.com/revive-2018-ICST>
- **A4:** 08/2018 – *RESPIRE* at Digital Carnival 2018 with Mirjana Prpa and Philippe Pasquier Vancouver, BC, Canada – Role: Artist & Developer – Link: <https://kivanctatar.com/respire>
- **A5:** 04/2018 – *REVIVE* at CHI Connexions with Philippe Pasquier and Remy Siu at the Société des Arts Technologique (SAT) dome (157 speakers and 8 projectors), 7 performances as a part of CHI 2018 Conference on Human Factors in Computing Systems, Montreal, Quebec, Canada – Role: Performer & Developer – Link: <https://kivanctatar.com/revive-2018-CHI>
- **A6:** 04/2018 – *RESPIRE* at CHI Virtual Reality exhibiton with Mirjana Prpa and Philippe Pasquier, as a part of CHI 2018 Conference on Human Factors in Computing Systems, Montreal, Quebec, Canada – Role: Artist & Developer – Link: <https://kivanctatar.com/respire>
- **A7:** 04/2018 – *RESPIRE* at Vancouver Art Gallery with Mirjana Prpa and Philippe Pasquier as a part of the conference Museums and the Web MWX18, Vancouver, BC, Canada – Role: Artist & Developer – Link: <https://kivanctatar.com/respire>

- **A8:** 03/2018 – Eternal Pink Noise Machine, with Philippe Pasquier, @*Pink Noise Pop Up* Exhibition in Seoul, South Korea – Role: Artist & Developer – Link: <https://kivanctatar.com/eternal-pink-noise-machine>
- **A9:** 09/2017 – MASOM joins two media art companies from Istanbul, Ouchhh and AudioFil for a projection mapping piece on the Facade of the Bolshoi Theatre, at the Circle of Light 2017, Moscow, Russia – Role: Artist & Developer – Link: <https://kivanctatar.com/masom-factor-v1-03>
- **A10:** 09/2017 – MASOM joins two media art companies from Istanbul, Ouchhh and AudioFil for a projection mapping piece on the Facade of the Romanian Parliament, at the IMapp Bucharest 2017, Bucharest, Romania – Role: Artist & Developer – Link: <https://kivanctatar.com/masom-factor-v1-03>
- **A11:** 09/2017 *IOTA_AI*, MASOM joins two media art companies from Istanbul, Ouchhh and AudioFil for a performance at the Ars Electronica Festival 2017 with the theme Artificial Intelligence. The team performed three times at the Deep Space 8K during the festival. Linz, Austria – Role: Artist & Developer – Link: <https://kivanctatar.com/masom-factor-v1-03>
- **A12:** 06/2017 – *MA_Test SOM_Pattern*, with the project Patar, in the collective concert by *Musical Metacreation Concert* Atlanta, Georgia, USA – Role: Performer & Developer – Link: <https://kivanctatar.com/MASOM-0-06>
- **A13:** 04/2017 – *Patar @CoCreaTive*, in the collective concert *Barely Constrained* by Co.Crea.Tive Vancouver, BC, Canada – Role: Performer & Developer – Link: <https://kivanctatar.com/MASOM-0-06>
- **A14:** 02/2017 *A Big MASOM Family*, in the collective concert *RE-UN-SOLVED* by Co.Crea.Tive Vancouver, BC, Canada – Role: Performer & Developer – Link: <https://kivanctatar.com/MASOM-0-04>
- **A15:** 12/2016 – *Tatar and MASOM take the AID train*, in the collective concert *Take the AID Train* by A.I.D, İstanbul, Turkey – Role: Performer & Developer – Link: <https://kivanctatar.com/MASOM-0-03>
- **A16:** 12/2016 – *madMethod* by NOW Society, with Stefan Smulovitz, Sammy Chien, MASOM, Philippe Pasquier, Lisa Cay Miller, Jon Bentley, JP Carter, James Meger, Skye Brooks, at Orpheum Annex, Vancouver, BC, Canada – Role: Performer & Developer
- **A17:** 11/2016 – *Pulse.Breath.Water* – with Mirjana Prpa, Philippe Pasquier, Bernhard Reicke in the VR exhibition at MUTEK_IMG – Virtual Reality (head mounted display and

headphones), generative audio, embodied interaction (via breath sensors), Montreal, Quebec, Canada – Role: Artist & Developer – Link: <https://kivanctatar.com/Pulse-Breath-Water>

- **A18:** 10/2016 – *A Conversation with AI*, in the collective concert *Open to Enter* by CoCreate, Vancouver, BC, Canada – Role: Performer & Developer – Link: <https://kivanctatar.com/MASOM-0-01>
- **A19:** 09/2016 – *Musebot Chill-out Session*, with Arne Eigenfeldt, Matthew Horrigan, Paul Paroczai, Oliver Bown, Ben Carey, Toby Gifford, Jeffrey Morris, and Si Wait, Sound and Music Computing, SMC 2016, Hamburg, Germany – Role: Artist & Developer
- **A20:** 07/2016 – *P.O.E.M.A.* with Regina Miranda, Mirjana Prpa, Philippe Pasquier, and Bernhard Reicke; Generative Audio (quadrophonic setup), Choreographic Installation, Virtual Reality (head mounted display and projection), Embodied Interaction (via respiration sensors), at the gallery *Oi Futuro*, as a part of the cultural program at OLYMPICS 2016, Rio de Janeiro, Brazil – Role: Artist & Developer & Composer – Link: <https://kivanctatar.com/POEMA>
- **A21:** 07/2016 – *Musebot Chill-out Session*, with Arne Eigenfeldt, Matthew Horrigan, Paul Paroczai, Oliver Bown, Ben Carey, Toby Gifford, and Jeffrey Morris, International Conference on New Interfaces for Musical Expression, NIME 2016, Brisbane, Australia – – Role: Artist & Developer
- **A22:** 03/2016 – *Pulse.Breath.Water* - Mirjana Prpa - Kıvanç Tatar, Philippe Pasquier, and Bernhard Reicke, in the exhibition *Scores+Traces: exposing the body through computation* - Virtual Reality (head mounted display and headphones), generative audio, embodied interaction (via breath sensors) @*One Art Space*, New York, NY, USA – Role: Artist & Developer – Link: <https://kivanctatar.com/Pulse-Breath-Water>
- **A23:** 03/2016 – *Tuned Ocean no.2*, in the exhibition *Scores+Traces: exposing the body through computation* - sound installation, generative audio, @*One Art Space*, New York, NY, USA – Role: Artist & Developer
- **A24:** 2015 – *Musebot Chill-out Session*, with Arne Eigenfeldt, Matthew Horrigan, Paul Paroczai, Oliver Bown, Ben Carey, Toby Gifford, and Jeffrey Morris, Generative Art Conference 2016, Venice, Italy – Role: Artist & Developer

1.5 Structure of Thesis

In the below, we guide the reader for the remaining of this thesis while clarifying the contributions of co-authors. The structure of this thesis and the topics covered by chapters are as follows:

- **Chapter 2** starts by an introduction to Computational Creativity, Metacreation, and Musical Metacreation (Section 2.1 and 2.2). The chapter surveys 78 musical agent architectures (Table 2.1) that has been covered in peer-reviewed publications. Section 2.3 proposes a typology of musical agents. Section 2.4 2.5, and 2.6 groups the musical agents by their architecture type. Chapter 2 also includes a typology of evaluations methodologies for musical agent architectures in Section 2.7. Section 2.8 points out the possible future directions for musical agents research, clarifies the interdisciplinarity of MuMe, and proposes a revised typology of MuMe systems.

This survey is carried out and written by Kivanç Tatar while Philippe Pasquier supervised the project.

- **Chapter 3** introduces the main implementation of audio-based musical agents with unsupervised learning. The agent architecture is titled Musical Agent based on Self-Organizing Maps (MASOM). This chapter introduces the main motivations of MASOM architecture. The technical details of MASOM architecture is further clarified in Chapter 4; hence, the reader may pass Section 3.4 and refer to Chapter 4.

MASOM is implemented and written by Kivanç Tatar while Philippe Pasquier supervised the project. The examples and documentation of public performances are available at <https://kivanctatar.com/masom>.

- **Chapter 4** delves deeper into the MASOM architectures by clarifying all variations. The chapter covers four statistical sequence modelling algorithms that have been tested within the MASOM architecture. These algorithms are Factor Oracle, Recurrent Neural Networks, Variable Markov Model Max-Order variation, and Variable Markov Model Prediction by Partial Matching C variation. This chapter evaluates the MASOM architecture by comparing these four algorithms using analytic measures. The test cases of the evaluation focus on two corpora of electroacoustic music and experimental electronic music with repetition.

This chapter presents the evaluation of the MASOM architecture. The contributors to this research paper include the first author Kivanç Tatar who introduced the hypothesis, implemented the algorithm, leaded the project, and wrote 90% of the paper, In collaboration with Kivanç Tatar, Jeff Ens provided the evaluation methodologies, Jonas Krasch implemented the Variable Markov Model Max-Order variation, Jianyu Fan run the multivariate regression for music emotion recognition model. Philippe Pasquier provided the project supervision.

- **Chapter 5** introduces the audio-visual performance project *Revive*. The project includes three sonic performers: two human musicians, and MASOM. This chapter exemplifies how MASOM architecture can be deployed to a musical context. The performance medium is a projection on a dome structure with multichannel 3D audio. The generative visuals in Revive are audio-reactive, and each performer has corresponding visuals so that the sonic gestures are

further emphasized using visuals. *Revive* incorporates various 3D audio spatialization methods including generative trajectories, and a cue system to automatize the performance cues. Each cue constraints the performers in a different way, and these constraints result in a performance of structured improvisation.

This chapter exemplifies how MASOM architecture can be incorporated into an audio-visual performance. The implementation of MASOM architecture, 3D audio spatialization, machine listening modules, and performance cue system is carried out by Kivanç Tatar. Remy Siu and Kivanç Tatar created the audio-reactive visual engine. The paper is written by Kivanç Tatar, and Philippe Pasquier provided the supervision the project and paper. The documentation is available at <https://kivanctatar.com/revive>.

- **Chapter 6** is the second application of audio-based musical agents. This application implements the musical agent within a Virtual Reality artwork, *Respire*. The artwork aims to connect the user with the virtual environment through the respiration patterns of the user. The musical agent perceives the patterns in the respiration, and maps these patters to the eventfulness of the audio samples that are used to produce generative music in the piece. The machine listening module calculated eventfulness and pleasantness of audio samples in advance, using a sound affect estimation module based on multivariate regression.

This chapter presents a Virtual Reality application of an audio-based musical agent architecture. The contributors to this research paper include first author Kivanç Tatar who implemented the musical agent architecture and breathing signal analysis, and wrote 90% of the Leonardo Music Journal paper, Mirjana Prpa who developed the virtual environment and wrote the remaining 10% of the Leonardo Music Journal paper, and Philippe Pasquier who carried out the project supervision. The documentation is available at <https://kivanctatar.com/respire>.

- **Chapter 7** concludes this thesis while pointing the future research directions of audio-based musical agents with unsupervised learning.
- **Appendices** introduce the related research projects and collaborations that are carried out during the doctoral studies.
- The source code of the main implementation of this thesis can be found in <https://github.com/ktatar>

1.6 Summary

In this first chapter, we started our introduction by emphasizing the relation between Art and Technology. Our introduction continued with the background knowledge on Generative Art and Computational Creativity. We mentioned the definitions of Metacreation, Musical

Metacreation, and Musical agents while clarifying our understanding of music that is in parallel with the theories of experimental electronic music. We explained the motivations behind musical agent architectures based on Self-Organizing Maps for audio applications, and we listed the research questions in focus. We clarified the contributions of this thesis to the literature while listing all publications and artworks that are outcomes of this thesis. Lastly, Section 3.4 guides the reader for remaining chapters in this thesis.

Bibliography

- [1] *Oxford Dictionaries English*. Oxford University Press, 2017. URL <https://en.oxforddictionaries.com/definition/style>.
- [2] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. Omax brothers: a dynamic topology of agents for improvisation learning. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 125–132. ACM Press, 2006. URL <http://dl.acm.org/citation.cfm?id=1178742>.
- [3] Margaret A. Boden. Creativity and ALife. *Artificial Life*, 21(3):354–365, August 2015. ISSN 1064-5462, 1530-9185. doi: 10.1162/ARTL_a_00176. URL http://www.mitpressjournals.org/doi/10.1162/ARTL_a_00176.
- [4] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep Learning Techniques for Music Generation-A Survey. *arXiv preprint arXiv:1709.01620*, 2017.
- [5] Bruce G Buchanan. Creativity at the Metalevel AAAI-2000 Presidential Address. *AI Magazine*, 22(3):16, 2001.
- [6] Joel Chadabe. *Electric sound: the past and promise of electronic music*. Prentice Hall, Upper Saddle River, N.J, 1997. ISBN 978-0-13-303231-4.
- [7] Nick Collins. Towards Machine Musicians Who Have Listened to More Music Than Us: Audio Database-Led Algorithmic Criticism for Automatic Composition and Live Concert Systems. *Computers in Entertainment*, 14(3):1–14, January 2017. ISSN 15443574. doi: 10.1145/2967510. URL <http://dl.acm.org/citation.cfm?doid=3023312.2967510>.
- [8] Simon Colton and Geraint A. Wiggins. Computational Creativity: The Final Frontier? *Frontiers in Artificial Intelligence and Applications*, pages 21–26, 2012. ISSN 0922-6389. doi: 10.3233/978-1-61499-098-7-21. URL <http://www.medra.org/servlet/aliasResolver?alias=iospressISSNISBN&issn=0922-6389&volume=242&spage=21>.
- [9] Roger B. Dannenberg. Style in Music. In Shlomo Argamon, Kevin Burns, and Shlomo Dubnov, editors, *The Structure of Style*, pages 45–57. Springer Berlin Heidelberg, 2010. ISBN

978-3-642-12336-8 978-3-642-12337-5. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-642-12337-5_3. DOI: 10.1007/978-3-642-12337-5_3.

- [10] Shlomo Dubnov, G. Assayag, and A. Cont. Audio Oracle Analysis of Musical Information Rate. In *Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC)*, pages 567–571, September 2011. doi: 10.1109/ICSC.2011.106.
- [11] Philip Galanter. What is generative art? Complexity theory as a context for art theory. In *Proceedings of the 6th Generative Art Conference*, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.2634>.
- [12] Benjamin Lévy, Georges Bloch, and Gérard Assayag. OMaxist dialectics. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 137–140, 2012. URL <https://hal.archives-ouvertes.fr/hal-00706662/>.
- [13] Russolo Luigi. *The Art of Noise*. A Great Bear Pamphlet, 1967.
- [14] Michael F. Lynch. Motivation, Microdrives and Microgoals in Mockingbird. In *Proceedings of 3rd International Workshop on Musical Metacreation (MUME 2014)*, North Carolina, USA, 2014. URL <http://musicalmetacreation.org/mume2014/content/proceedings/Motivation,%20Microdrives%20and%20Microgoals%20in%20Mockingbird.pdf>.
- [15] Jérôme Nika, Marc Chemillier, and Gérard Assayag. ImprotoK: Introducing Scenarios into Human-Computer Music Improvisation. *Computers in Entertainment*, 14(2):1–27, January 2017. ISSN 15443574. doi: 10.1145/3022635. URL <http://dl.acm.org/citation.cfm?doid=3023311.3022635>.
- [16] Doug Van Nort, Pauline Oliveros, and Jonas Braasch. Electro/Acoustic Improvisation and Deeply Listening Machines. *Journal of New Music Research*, 42(4):303–324, December 2013. ISSN 0929-8215. doi: 10.1080/09298215.2013.860465. URL <http://dx.doi.org/10.1080/09298215.2013.860465>.
- [17] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An Introduction to Musical Metacreation. *Computers in Entertainment*, 14(2):1–14, January 2017. ISSN 15443574. doi: 10.1145/2930672. URL <http://dl.acm.org/citation.cfm?doid=3023311.2930672>.
- [18] Curtis Roads. *Microsound*. The MIT Press, Cambridge, Mass., August 2004. ISBN 9780262681544.
- [19] Curtis Roads. *Composing electronic music: a new aesthetic*. Oxford University Press, Oxford, 2015. ISBN 978-0-19-537324-0.

- [20] Martin Siefkes. Style: A new semiotic view on an old problem. *Kodikas/Code. Ars Semeiotica*, 34(1-2), 2011.
- [21] Denis Smalley. Spectromorphology: explaining sound-shapes. *Organised Sound*, 2(02):107–126, August 1997. ISSN 1469-8153. doi: 10.1017/S1355771897009059. URL http://journals.cambridge.org/article_S1355771897009059.
- [22] Karlheinz Stockhausen. Four Criteria of Electronic Music with Examples from Kontakte, 1972. URL <https://www.youtube.com/watch?v=7xyGtI7KKIY&list=PLRBdTyZ761vAFotZvocPjpRVTL6htJzoP>.
- [23] Bob L. Sturm, Oded Ben-Tal, Úna Monaghan, Nick Collins, Dorien Herremans, Elaine Chew, Gaëtan Hadjeres, Emmanuel Deruty, and François Pachet. Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, 48(1):36–55, January 2019. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1515233. URL <https://www.tandfonline.com/doi/full/10.1080/09298215.2018.1515233>.
- [24] Kivanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):1–50, September 2018. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1511736.
- [25] Nicolas Gonzalez Thomas, Philippe Pasquier, Arne Eigenfeldt, and James B. Maxwell. A Methodology for the Comparison of Melodic Generation Models Using Meta-Melo. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 561–566, Brazil, 2013. ISBN 978-0-615-90065-0. URL http://ismir2013.ismir.net/wp-content/uploads/2013/09/228_Paper.pdf.
- [26] Barry Truax. *Acoustic Communication*. Greenwood Publishing Group, 2001. ISBN 978-1-56750-536-8.
- [27] Edgard Varese and Chou Wen-chung. The liberation of Sound. *Perspectives of New Music*, 5(1):11–19, 1966. URL https://www.jstor.org/stable/832385?origin=JSTOR-pdf&seq=1#page_scan_tab_contents.
- [28] Cheng-I Wang, Jennifer Hsu, and Shlomo Dubnov. Machine Improvisation with Variable Markov Oracle: Toward Guided and Structured Improvisation. *Computers in Entertainment*, 14(3):1–18, January 2017. ISSN 15443574. doi: 10.1145/2905371. URL <http://dl.acm.org/citation.cfm?doid=3023312.2905371>.
- [29] Mitchell Whitelaw. *Metacreation: art and artificial life*. MIT Press, Cambridge, Mass, 2004. ISBN 9780262232340.
- [30] Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, June 2009. ISBN 9780470519462.

Chapter 2

Musical Agents: a Typology and State of the Art towards Musical Metacreation

KIVANÇ TATAR
PHILIPPE PASQUIER

AS PUBLISHED IN JOURNAL OF NEW MUSIC RESEARCH, 47(4), PP 1–50, 2018,
DOI.ORG/10.1080/09298215.2018.1511736

Abstract

Musical agents are artificial agents that tackle musical creative tasks, partially or completely. We frame our review of musical agents by combining the terminology of Generative Arts (artistic practice) and the scientific literature of Computational Creativity, Multi-agent Systems, and Artificial Intelligence. We define Musical Metacreation as a field that studies the partial or complete automation of musical tasks. Autonomy, reactivity, adaptability, coordination, emergence, and proactivity are the key concepts of Musical Metacreation and musical agents. We survey seventy-eight musical agent systems, and present a typology of musical agents based on the agent literature in Multi-agent systems and Artificial Intelligence. We continue by examining the evaluation methodologies of musical agents. We propose possible future steps while mentioning ongoing discussions in the field.

Keywords: Musical agents; Multi-Agent Systems; Artificial Intelligence; Musical Metacreation; Computational Creativity; Virtual Reality; Interactive Arts; Contemporary Arts

2.1 Introduction

The works of Generative Music rely on autonomous systems for part or all of their production. One type of such systems is the automaton, a self-operating machine that carries out pre-defined procedures. The first musical automaton, al-Jazari's water clock, appeared in the 7th century as a result of advances in hydraulics [84]. This water clock could generate music using a mechanical and hydraulic system. With the onset of the industrial revolution and following the invention of electricity, automatic musical machines entered a new phase in their development, and more musical automata emerged. These included amongst other the Regina Concert Orchestriion, the Autophone, and the Link Orchestriion. And now, after the digital revolution, artificial agents have become the modern day equivalent of the automaton.

In the digital age, new autonomous tools and systems have been emerging in creative applications. Artificial Intelligence (AI) and Multi-Agent Systems (MAS) are two examples of fields that provide such autonomous tools. Simon [168] defines AI as 'the science of having machines solve problems that do require intelligence when solved by humans. MAS are distributed/concurrent systems that are autonomous, able to make independent decisions, and run online [201]. Software agents in MAS are autonomous pieces of software which contain perception and action abilities. Applications of MAS are beneficial to modelling and designing musical creativity because musical creativity involves distributed, coordinated entities with perception and action abilities. For example, in a live music performance, musicians collaboratively create music by listening to each other. Similarly, we could distribute composition tasks into such sub-tasks as producing individual instrument parts or layers in experimental music [163].

Our review gives an introduction to researchers and practitioners who are interested in musical agents. Musical agents are artificial agents that tackle musical creative tasks, in part or as a whole, and use the methods of MAS and Artificial Intelligence to automatize these tasks. Thus, this topic is naturally interdisciplinary, combining music, science, design, and technology. In this paper, we present a state of the art in musical agents that utilize MAS technologies for musical creativity. Three types of artificial agents appear in the literature: software agents that are purely computational; virtual agents that are embodied in a Computer Generated Image (CGI)¹; and robotic agents that hold a physical form². In our survey of Musical Agents, we focus on purely computational software agents, and exclude the virtual agent applications in CGI, and robotics.

More specifically, the survey covers seventy-eight musical agent systems compiled in Table 2.1. Certainly, many more musical agents have been developed within the artistic practices. However, we only cover the systems whose details are given in peer-reviewed publications. The systems are referenced throughout the paper using the name convention *system-name* (#), where the circled number refers to the system numbers in Table 2.1. We propose a taxonomy (Figure 2.2) that is framed us-

¹Please refer to Cassell et al. [46] and Hartholt et al. [100] for an introduction to virtual agents in CGI.

²Bretan and Weinberg [38] present a state of the art in musically creative robotic agents.

ing the terminology of MAS, AI, and Computational Creativity (CC). We incorporated established dimensions and categorizations of these fields in our taxonomy rather than coming up with new ones. We aimed for a terminology that is inclusive of both Generative Music (an artistic practice) and Computational Creativity for Music (a scientific research field). In the next section, we supply a background of these associated fields. We present a typology of musical agents, and extend the agent classification of MAS to include the particularities of musical agents and introduce the various dimensions of musical agents in Section 2.3. We subsequently group musical agents according to their MAS architecture, and present details on each system in Section 2.4, 2.5, and 2.6. Then, we discuss the evaluation of musical agents in Section 2.7. In the last section, we propose Musical Metacreation as a field that combines science (Computational Creativity) and artistic practice of Generative Music; and propose possible future directions in the field.

2.2 Generative Art and Computational Creativity

We build our review using the terminologies of Generative Art, an artistic practice, and Computational Creativity, a scientific field. Before we start the survey of musical agents, we would like to introduce the fields that encompass musical agents. We first make a note of two generic fields, Generative Art and Computational Creativity, then we continue to more specific fields that are Metacreation and Musical Metacreation.

The roots of Computational Creativity can be traced back to Generative Art as well as AI, Artificial Life (A-Life), Machine Learning, and Cognitive Sciences. Galanter [88] defines Generative Art as follows:

Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.

We observe in this review that some musical agent systems inherit rules that are strictly defined by their authors whereas other systems adapt their aesthetics by a learning process. That is, the degree of genericity varies in autonomous systems as well as musical agents. The genericity of musical agents thus spans a continuous dimension that ranges from specific systems to purely generic systems. Many rule-based musical agents lean towards the specific end of the genericity continuum. For example, *Voyager* (11) includes 15 pitch generation algorithms that are strictly defined by its creator, George Lewis [116], as such strictly implements the aesthetics of its creator. In comparison, the *Continuator* (50) can learn the style of any musician and does not include pre-defined music rules. The Voyager is closer to the specific end of this continuum whereas the Continuator stands closer to the generic end.

The notion of autonomy frequently emerges when we discuss generative systems. As Galanter's definition emphasizes, all generative systems possess a degree of autonomy. Hence, we define a di-



Figure 2.1: The continuum of autonomy

dimension of autonomy that is continuous, ranging from purely reactive systems without autonomy to completely autonomous systems (Figure 2.1). For example, MuseScore³ is a music notation software. MuseScore only produces music as a direct result of the user's input and is purely reactive. In contrast, the *Continuator* [50] autonomously learns from its user's input and continues a melody when the user/musician stops playing. The computer assisted creativity would fall in the middle range of the autonomy continuum. An example of computer assisted creativity is assisted composition in music (see Section 2.3).

Autonomous systems of Generative Art focus on artistic creative tasks. Note that, there are also creative tasks that are not artistic. For example, creating a culinary recipe [3] is a creative task that is not artistic. The academic field, Computational Creativity studies computational processes for all creative tasks including the artistic ones. Creative tasks of art and music are different than problems with optimal solutions. In the case of problems with optimal solutions, the quality measures are well-defined. For example, we evaluate the performance of a software agent that aims to optimize fuel consumption, by measuring the actual fuel usage. In contrast, creative tasks of art and music tackle problems that lack definitive or optimal solutions. The solutions of creative tasks of art and music have ill-defined quality measures, e.g., there is no notion of optimal music, nor universal measure of quality in musical improvisation. Colton and Wiggins [53] define Computational Creativity as,

The philosophy, science and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviours that unbiased observers would deem to be creative.

The research in Computational Creativity mainly centres around the following three common themes:

- Computational models of (human) creativity: Such studies research creativity using computational models. In the case of artificial agents, the possibilities can also go beyond human capabilities. For example, Collins [51] proposes the notion of a musical agent that listens to more music than humans could.

³<https://musescore.org/>

- Computational systems for supporting creativity: These systems are smart assistants for creative applications. These assistants can suggest solutions and alternatives to the user by analysing the user’s behaviour. Musical agents focusing on assisted composition are examples of such systems.
- Artificial creative systems: Computational models of creativity are studied through the development of artificial creative systems. We propose to define these systems as Metacreations.

We call Metacreation the domain that both study and produce systems that partially or completely automate creative tasks. The notion of Metacreation and Meta-level creativity was mentioned by many [43, 199, 200, 28] and the term was explicitly proposed by Whitelaw [198]. The term also resonates back to the artistic statements (by artists such as Nicholas Schöffer and James Seawright) in 1950s and 1960s [198].

There are two types of creativity that Metacreation explores. First, the simulation of human creativity is *creativity as it is*. For instance, *Continuator* (50) is a musical agent that implements *musical creativity as it is*, by imitating the musical style of a performer. Second, exploring creative processes that humans are incapable of, is *creativity as it could be*. For example, *Shoals* (44) explores *musical creativity as it could be*, by sonifying the actions of a virtual ant colony.

Building on this literature, Pasquier et al. [155] define *Musical Metacreation* as “...a subfield of Computational Creativity that addresses music-related creative tasks.” We revisit this definition and we propose that *Musical Metacreation* is the partial or complete automation of musical tasks. MuMe, as an interdisciplinary field, is inclusive of all approaches, studies, domains, and practices that automatize musical tasks. We acknowledge that several other domains also study the topics of MuMe, and we elaborate on this in Section 2.8.4. We propose to define MuMe as a field that uses the terminology of Generative Art (practice) and Computational Creativity (science) to cover autonomous systems of algorithmic music, generative music, machine musicianship, and machine improvisation. The applications of MuMe use techniques of computational models, Artificial Intelligence (AI), and MAS to automatize musical tasks. Musical agents is a sub-category of MuMe, and in the next section, we delve into musical agents and the musical tasks that are carried out by musical agents.

2.3 Typology of Musical Agents

A musical agent is an artificial agent that partially or completely automates musical creative tasks. In the following, we explain the terms ‘musical’ and ‘agent’. We refer to the term musical in the context of Varése’s *Organized Sound* [189]. In this survey, the definition of music is inclusive of all works that use sound as a medium. Hence, we also include implementations of Sound Art, Sonic Arts, and Contemporary Art works using the sound medium.

Although there is no consensus on the definition of agents in Social Sciences and Philosophy [81], an agent is a well-defined term in Computer Sciences. An agent is an autonomous system that initiates actions to respond to its environment in timely fashion [201]. Similarly, musical agents

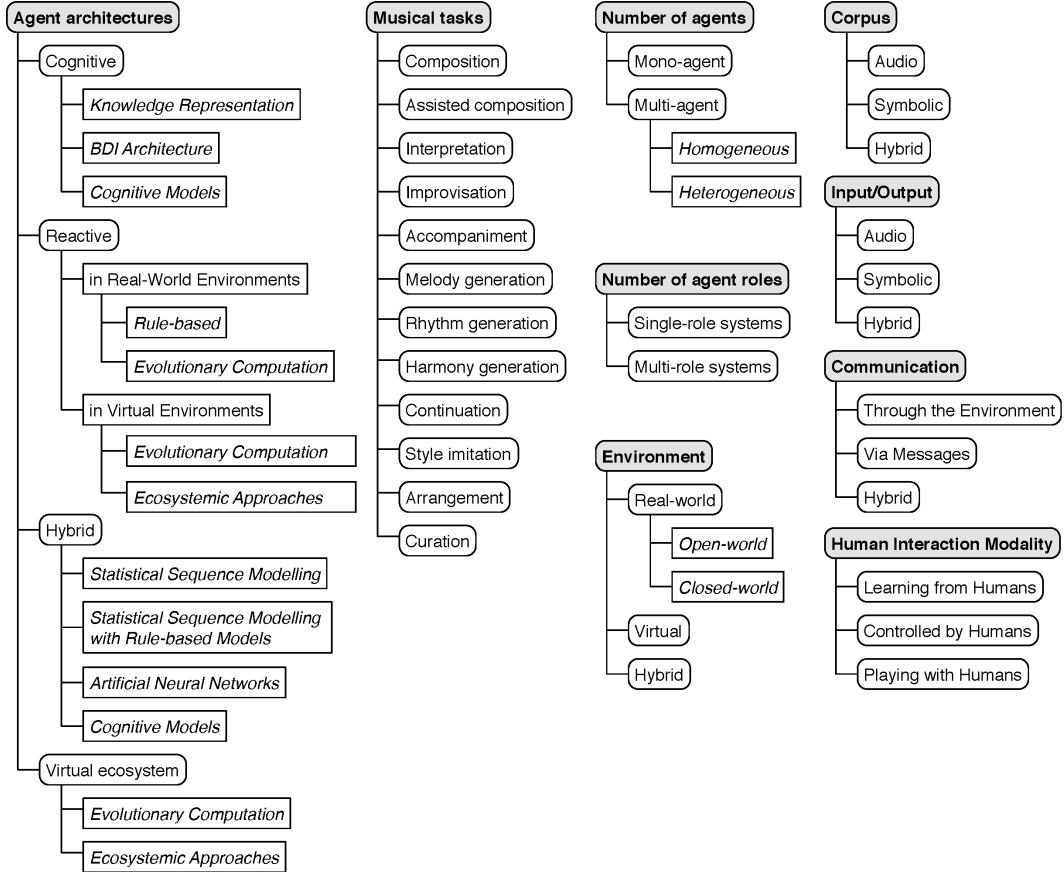


Figure 2.2: The 9 dimensions of our musical agents typology

explore the notions of autonomy, reactivity, proactivity, adaptability, coordination, and emergence. In this survey, we include musical agents that implement communication but do not implement machine listening. However, we exclude musical agents that solely analyse music, and purely generative systems⁴ that have neither perception capabilities nor communication abilities. Some musical agents work offline such as *MASC* (77) while others work online such as the *Contuniator* (50) and the *Voyager* (11). There are also musical agents that learn offline and generate online such as *MASOM* (78).

There is a wide variety of musical agents. We reviewed 78 systems and identified 9 dimensions that form the typology of musical agents. These 9 dimensions are agent architectures, musical tasks, environment types, number of agents, number of agent roles, communication types, corpus types, input/output (I/O) types, human interaction modality (HIM). This typology is available in Figure 2.2 and Table 2.1.

⁴Herremans et al. [102] recently surveyed purely generative systems from a conventional music perspective.

1. **Agent architectures:** Our typology of musical agent architectures (Figure 2.2) is based on well-known agent classifications in Multi-Agent Systems and Artificial Intelligence literature. On the top level of the musical agent architecture typology, we classify musical agent architectures using three broad types of agent architectures that are proposed in MAS: cognitive, reactive, and hybrid [201, 166, 196]. Under the agent types, we use architecture model paradigms as another level of categorization. This classification of agent architectures and model paradigms also serves as the base along which we discuss our survey of musical agents, and the details on each agent architecture type are given in the corresponding sections.
2. **Musical tasks:** Musical agents partially or completely automatize musical creative tasks. So far, we identified twelve different musical tasks implemented by musical agents (Figure 2.2):
 - **Composition:** The artefacts of composition are sets of symbolic instructions in the case of musical scores, or audio files in the case of fixed-media works in electroacoustic music or acousmatic music. For example, *Coming Together:Freesound* (25) is a system that generates soundscape compositions.
 - **Assisted composition** systems recommend musical ideas to composers by automatizing any sub-tasks of musical composition. For example, *MASC* (60) implements Affective Computing with a MAS to recommend melodies to composers. Also, several composers used *OMAX* (54) to recombine and transform musical material for composition tasks ⁵.
 - **Interpretation:** Performers interpret a set of musical instructions to produce sounds or generate audio, which we refer to as interpretation tasks. For example, *IMAP* (36) evolves different interpretations of the same musical phrase using a MAS. Interpretation tasks can also appear in the musical tasks of symbolic (notated) music.
 - **Improvisation:** We can break down the improvisation task into real-time distributed composition and real-time interpretation tasks. For example, *MASOM* (78) performs free improvisation with or without software and human agents in the context of experimental electronic music. Moreover, musical agent improvisation is also referred to as machine improvisation.
 - **Accompaniment** tasks incorporate following and supporting a leading performer or musical part. For example, *Virtualband* (15) follows the eventfulness of a performer’s audio and generates rhythm, chord progressions, and bass parts. Accompaniment task can appear in composition, interpretation, and improvisation tasks.
 - **Melody, rhythm, and harmony generation** tasks appear as sub-tasks of composition, assisted composition, interpretation, and improvisation.

⁵Musical examples of OMAX in practice can be found at <http://repmus.ircam.fr/omax/home>.

- **Continuation** consists of having a musician play or improvise, and the system taking over once the musician stops. For instance, the *Continuator* (50) is a musical agent that carries on a musical phrase played by a human performer in the style of the human performer.
- **Style imitation:** Given a corpus $C = C_1, \dots, C_n$ representative of style S , style imitation is to generate new instances that would be classified as belonging to S by an unbiased observer (typically a set of human subjects). For example, the *Audio Oracle* (58) is a musical agent that uses machine listening to imitate the style of another performer.
- **Arrangement:** The selection and temporal ordering of musical material are the main tasks of musical arrangement. We differ arrangement from instrumentation which is the assignment of parts of the music to specific musical instruments. In the case of musical agents, we encountered only one system, the fourth version of *Coming Together* (25) that implements arrangement.
- **Curation** differs from arrangement. Curation is the selection of agents to perform whereas arrangement is selecting and ordering musical material. For example, *ParamBOT* (27) is a musical agent that curates a selection of musical agents.

These musical tasks are neither mutually exclusive nor independent. For instance, a composition task may include sub-tasks of melody, rhythm, and harmony generation as in the case of *Inmamusys* (2). In contrast, *Coming Together:Freesound* (24) also implements composition tasks, and yet, it does not include any of the sub-tasks of melody, rhythm, and harmony generation.

3. **Environment types:** The literature considers three types of environments in musical agent systems. First, the *real-world* environment is the sound medium where an agent listens to the sum of all sounds generated by all agents. For example, in a duo setting, *Voyager* (11) listens to the human performer and outputs audio to the real-world environment so that the human performer can hear. There are three types of agents in real-world environments: physical, visual, and sonic. Physical agents in real-worlds are musical robots, and visual agents apply visualization of artificial agents. We mentioned that we do not cover robotic and CGI agents in this survey and we only cover sonic software agents that listen to either audio or symbolic music input. Furthermore, there are two sub-categories of real-world environments: *open-world* and *close-world*. Open-world environments allow human or software agents to enter and leave the environment during the generation stage whereas close-world environments do not. For example, *Voyager* (11), *Odessa* (16), and *MASOM* (78) listen to the real-world and allow human or software agents to enter and leave the environment.

Second, simulations of physical environments are *virtual environments*. In the literature, musical agent architectures utilize a virtual environment in three ways to generate audio: the virtual location of an agent, the spatial interactions between agents in the virtual space, and an agent’s interaction with the virtual environment such as finding virtual foods. For instance,

the agents in *Shoals*' (44) are situated in a virtual environment where they consume virtual foods. The system generates audio by sonifying the consumption of food. Also, the agents can create groups by communication through spatial interactions in *Shoals*. Third, the real world environment affects the virtual ecosystem in hybrid environments. For example, the video input in *Petri* (38) generates attraction points in the virtual environment. The agents try to move towards these attraction points in the virtual environment and the location of agents create the audio output. Moreover, the following properties of MAS environments are also applicable to musical agent environments [166]:

- *Fully observable vs. partially observable*: An environment is fully observable if an agent can perceive the environmental properties that are relevant to the choice of action. An environment is partially observable if an agent has no capabilities to perceive. For example, an agent in the system (35) perceives only one agent at a time although there are multiple agents in the system.
- *Deterministic vs. stochastic*: An environment is deterministic if the next state of the environment only depends on the previous state of the environment and the actions of the agents in the environment, such as the environment of *Beatbender* (20). Non-deterministic and partially observable environments are called stochastic environments. There are two types of stochastic environments: stationary and non-stationary. In stationary stochastic environments, there is only one stochastic model and the model does not change. For example, the probability distributions in *Virtualband* (15) does not change during the performance. In non-stationary stochastic environments, the stochastic model changes. The *Continuator* (50) is an example of an agent in non-stationary environment because the agent's stochastic model, that is the Markov model, changes continuously.
- *Episodic vs. sequential*: During each episode, an agent has a percept input and generates an action output. In an episodic environment, the current episode of an environment is independent of the previous episodes. For example, each time *GenJam* (28) starts playing a chorus, the solo is independent of the previous choruses' solo. In a sequential environment, the current episode depends on the previous episodes. For instance, the musical agent applications with the first or higher order Markov Models have sequential environments.
- *Static vs. dynamic*: An environment is static if it does not change while an agent is deliberating, else it is dynamic. An example of dynamic environment is the virtual environment in *Petri* (38) because the attraction points in the virtual environment change independently.
- *Discrete vs. continuous*: An environment is discrete if it has a finite number of distinctive states, and continuous if the environment has an infinite number of distinctive states. The applications with symbolic music representation have discrete environments when the parameters are discrete, such as pitch values in MIDI. In comparison, an audio environment is continuous.

4. **The number of agents:** We group musical agent systems in two categories with reference to the number of agents included: *mono-agent* and *multi-agent*. Mono-agent systems include only one musical agent whereas Multi-agent systems have many. Although we approach human performers as agents, we only include software agents in this categorization. Also, we present the interactions of musical agent systems with humans in the Human Interaction Modality (HIM) dimension. Human performers can play with a mono-agent system. For example, the *Continuator* (50) is a mono-agent system in a duo setting with a human performer. Multi-agent systems in which all the agents share the same architecture are said to be a *homogeneous* Multi-agent systems. For example, *MASOM* (78) has a flexible homogeneous architecture that allows the user to start more than one *MASOM* agent for a live performance. A musical agent system is a *heterogeneous* Multi-agent system if there are agents with different architectures. For example, *Cypher* (10) is a heterogeneous Multi-agent system with multiple agent architectures.
 5. **The number of agent roles:** All agents focus on the same task in *single-role* systems. In comparison, there are different roles that agents can take in *multi-role* musical agent systems. For example, there is only one type of agent architecture in *iME* (48), and agents can take the roles of either *listener* or *player* in an episode. Agents in *iME* can change their roles every episode.
 6. **Communication types:** There are three types of communication in musical agent systems: *through the environment*, *via messages*, and *hybrid*. First, through the environment communication is related to the notion of *stigmergy*. Stigmergy is the indirect coordination through the environment [103]. That is, an action of an agent leaves footprints in the environment. These footprints stimulate the agents in the environment. For example, ants leave traces when they find food in the environment [173]. Other ants follow these traces to reach the food. Similarly, musical agents implement machine listening of the real-world to communicate through the environment. For example, let's assume that a musical agent desires the ensemble to play louder. An agent expresses this desire by playing louder sounds instead of sending symbolic messages to the other agents in the ensemble. The other agents interpret this desire by listening to the real-world environment.
- Second, agents that communicate via messages use pre-defined, system specific messages. The developers of musical agents come up with protocols that specify the type of messages, and how agents send and receive messages. For example, the agents in *Inmamusys* (2) communicate via messages to generate compositions as notated symbolic music.
- Third, both of these methods are used in the *hybrid* communication. For instance, *Indifference Engine* (5) communicates through environment by listening to a human performer and generating audio, while the agents in the system communicates through system specific messages such as *pitch*, *volume*, *speed*, *tensionCurve*, and *confidence*.

In MAS, the communication between agents includes negotiation, bargaining, and argumentation [196]. An agent’s behavior can be cooperative or competitive. For example, playing chess is a competitive behavior whereas distributed problem solving is cooperative. Musical agent implementations are mostly cooperative given that the nature of making music is cooperative. Nevertheless, there are also examples of systems with competitive agents. For example, agents create social groups in *Shoals* (44), and these groups compete with each other to find and consume food in a virtual ecosystem.

7. **Corpus types:** A corpus is a set of symbolic music or audio samples in the case of musical agents. Musical agents use a corpus as a musical memory and knowledge. We group the corpus types of musical agents in three categories: symbolic, audio, and hybrid. A symbolic corpus is a set of symbolic representations of music. In most of the cases, the symbolic representation uses MIDI. An audio corpus is a set of audio samples. The samples in an audio corpus can range from grains of audio samples to full length music pieces. A hybrid corpus include a set of audio files with any kind of symbolic data. For example, *MASOM* (78) includes a hybrid corpus, that is, a set of audio segments with 18 dimensional feature vectors.
8. **Input/Output types:** We differentiate corpus types from I/O types. Corpus types are related to the learning and generation. However, the I/O types clarify how an agent listens and outputs to the environment. We observed three types of I/O in musical agents: *audio*, *symbolic*, or *hybrid*. The audio I/O is the audio signal perceived and generated by the agents. The agents with the symbolic I/O use a symbolic representation of music, that is, in most cases the MIDI protocol. Agents with the hybrid I/O use both audio and symbolic I/O.
9. **HIM:** Three types of Human Interaction Modality (HIM) emerge in musical agents: *systems learning from humans*, *systems controlled by humans*, and *systems playing with humans*. Some musical agent systems learn the style of a human performer or composer as in the case of the *Continuator* (50) and *MASOM* (78). Some systems include global variables that can be controlled by humans as in the case of *Kinectic Engine* (21) and *HARP* (71). Many of the musical agents, such as the systems focusing on machine improvisation in Section 2.6.1, can perform with human performers.

In the following sections, we define the agent architectures, group musical agents by their architecture type, and give details about each system.

2.4 Cognitive Musical Agents

Gomila and Müller [98] define a cognitive system as one that ‘learns from individual experience and uses its knowledge in a flexible manner to achieve its goals.’ Six aspects of cognitive systems are dealing with an uncertain world, learning from experience, understanding knowledge, flexible use

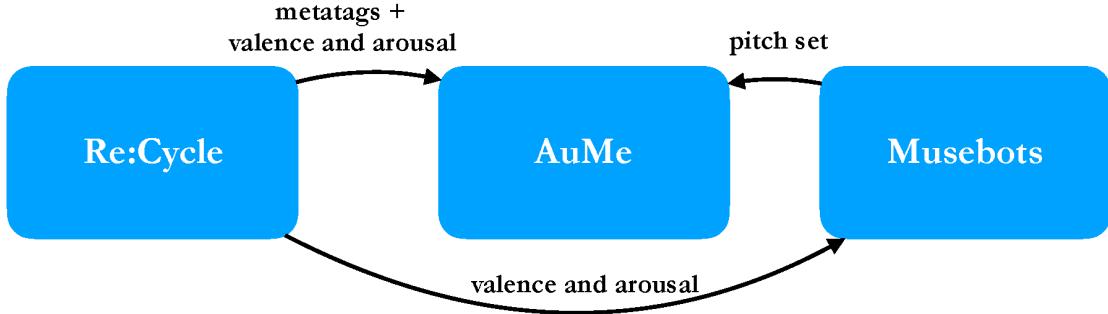


Figure 2.3: The block diagram of the system ③ in Table 2.1 [24]

of knowledge, autonomy, and social abilities. Cognitive agents inherit the properties of both cognitive systems and MAS. Cognitive models, in comparison, are software architectures that specifically model human cognitive processes. Some examples of cognitive models proposed in Cognitive Science are Ymir, ACT-R, Soar, NARS, OSCAR, AKIRA, CLARION, LIDA, and Ikon Flux [184]. We study cognitive musical agents in three categories in the following sections.

2.4.1 Cognitive musical agents with knowledge representation

A classic paradigm for cognitive agent architectures is *Logic-based agents* (LBAs) [201, 196]. LBAs perceive their environment by building and maintaining symbolic representations. The environment is represented as a set of assertions. LBAs reason about the environment using a set of logical rules and apply theorem-proving using the knowledge.

A recurrent theme in cognitive musical agents is the implementation of knowledge representation with a rule-based agent architecture. Inspired by music theory, the authors come up with a knowledge representation, and rules of their musical system. Alternatively, the agents generate logical assertions and rules using a corpus. In the following, we survey three systems with knowledge representations.

Wulffhorst et al. [202] studied fifty popular songs to devise a table of possible harmonic transitions. Vicari et al. [190] continue this work by presenting a Multi-agent system with multiple agent models, called Virtual Musical Multi-agent system, *VMMAS* ①. The application of this system is online music accompaniment. Seven types of agent models appear in *VMMAS* ①: Cautious (activates only when the agent has a high metric and harmonic confidence degree), Leader (simulates other agents), Flexible (adapts to the metric changes), Inflexible (does not adapt to the metric changes), Persuasive (tries to stabilize around its ‘ideal’ tempo), Improvising (proposes harmony progressions), and Lyric (proposes tempo changes). The authors state that agent models use ‘fuzzy cognition’; however, the authors have concealed how the fuzzy logic is implemented.

The second system, *Inmamusys* ② concentrates on a two-layer multi-agent system [58]. All agents in *Inmamusys* include a knowledge representation. In the first layer, *Inmamusys* ② chooses a composer agent that decides the number of voices, the timbre of the voices, tonality, and the number of measures. In the second layer, there are four types of agent models: melody, harmony,

accompaniment, and drums. The system allows a degree of human control through a graphical user interface (GUI). In this interface, the user can choose the desired emotion, instruments and duration of the composition. However, it is not disclosed how desired emotions are implemented.

The third system is a generative multimedia system ③ with affective computing [24]. This system generates soundscape, moving images and music. The affect estimation uses Russell's [165] two dimensional (valence and arousal) circumplex affect model. The system design includes three modules: Re:Cycle, AuMe (Audio Metaphor) and Musebots (see Figure 2.3). The Re:Cycle uses a corpus of moving images with valence and arousal tags. Re:Cycle module sends the desired valence and arousal values to AuMe and Musebots. Using a machine learning model (multivariate regression) for affect estimation in soundscape recordings, AuMe chooses soundscape recordings with the desired valence and arousal values. The Musebots module maps valence (pleasantness) and arousal (eventfulness) to multiple musical parameters such as musical consonance, melodic movements, and rhythm.

2.4.2 Cognitive musical agents with BDI architecture

One of the most common cognitive agent architectures is the Belief-Desire-Intention (BDI) architecture [201]. The BDI architecture⁶ applies *practical reasoning*. Practical reasoning is the goal-directed selection of actions. There are two aspects of practical reasoning: *deliberation* and *means-end reasoning*. Deliberation is the process of deciding what to achieve. The outputs of deliberation are intentions. Means-ends reasoning is how to achieve the goal that is set by the deliberation process. The result of means-ends reasoning is a *plan*.

An agent uses its beliefs to represent the state of the world and the know-how of the agent. Beliefs are different from the knowledge. Beliefs can be wrong and they are non monotonic. Internal or external perception update beliefs. Internal perception is the perception of the agent's own state whereas the external perception is the perception of the environment. Reasoning can also update the beliefs of an agent. There are two families of reasoning: perfect and bounded rationality. Perfect rationality is where we assume the logical omniscience of the agent. In comparison, bounded rationality is the notion of accepting the finite nature of the resources available for reasoning.

While beliefs are informational attitudes, the desires are motivational attitudes. Desires are not necessarily consistent or achievable. Deliberation is the process of choosing which desires are to be pursued according to the current beliefs. The agent generates intentions by applying a selection function to its desires.

Intentions are desires that the agent is committed to make happen. Agents determine ways to change the state of the environment so as to make the intentions true. Intentions provide a filter for the adaptation of the other intentions that must not conflict. Agents track the success of their intentions. Agents are inclined to try again if the attempts to satisfy an intention fail. That is, the

⁶Wooldridge [201] provides the psuedo code of the BDI agent control loop.

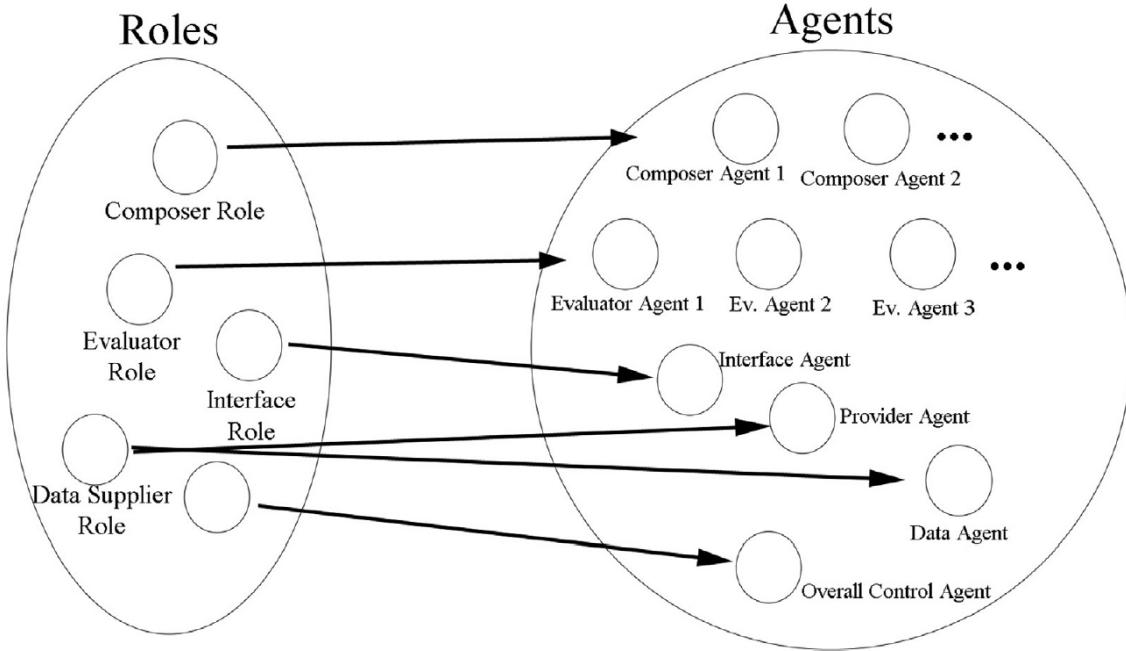


Figure 2.4: An example of role assessment in MUSIC-MAS [145]

intentions of agents are persistent. Agents believe that their intentions are possible and intentions are not closed under implications.

Means-end reasoning is to determine how the intentions are achieved. Agents generate a sequence of actions, that is, a plan. Plans can be deterministic and non-deterministic. Means-end reasoning generates or selects a plan to be executed as an attempt to achieve the intention.

We found three musical agent implementations that use BDI architecture explicitly. The first system ④, presented as a part of the Coming Together musical agent series, explores the idea of negotiation between musical agents [69]. The communication is hybrid, and through communication, the agents generate their own goals and create plans to achieve these goals. An agent in this system desires to create a repeating phrase that is updated by the communication between agents and the changes in the environment.

Indifference Engine ⑤ is the second musical agent application including BDI architecture with the hybrid communication [71]. Each agent generates an intention graph for *pitch*, *volume*, and *speed* at the beginning of performance. The average of these three graphs is the *tensionCurve* of an agent. Indifference Engine can run as a mono-agent system, multi-agent system, and multi-agent system including a human performer. The data of agents are globally shared; however, the intentions of agents are private. The agents also negotiate their intentions and argue on choosing a leader. Each agent follows the leader with a weight parameter called *confidence*. The confidence parameter is set by the proximity of the agent to the mean of other agents' pitch, volume, and speed parameters. The closer an agent is to the mean, the higher its confidence is. Moreover, the agents choose to either join the multi-agent ecosystem or follow the human performer. Also, the agents negotiate on which

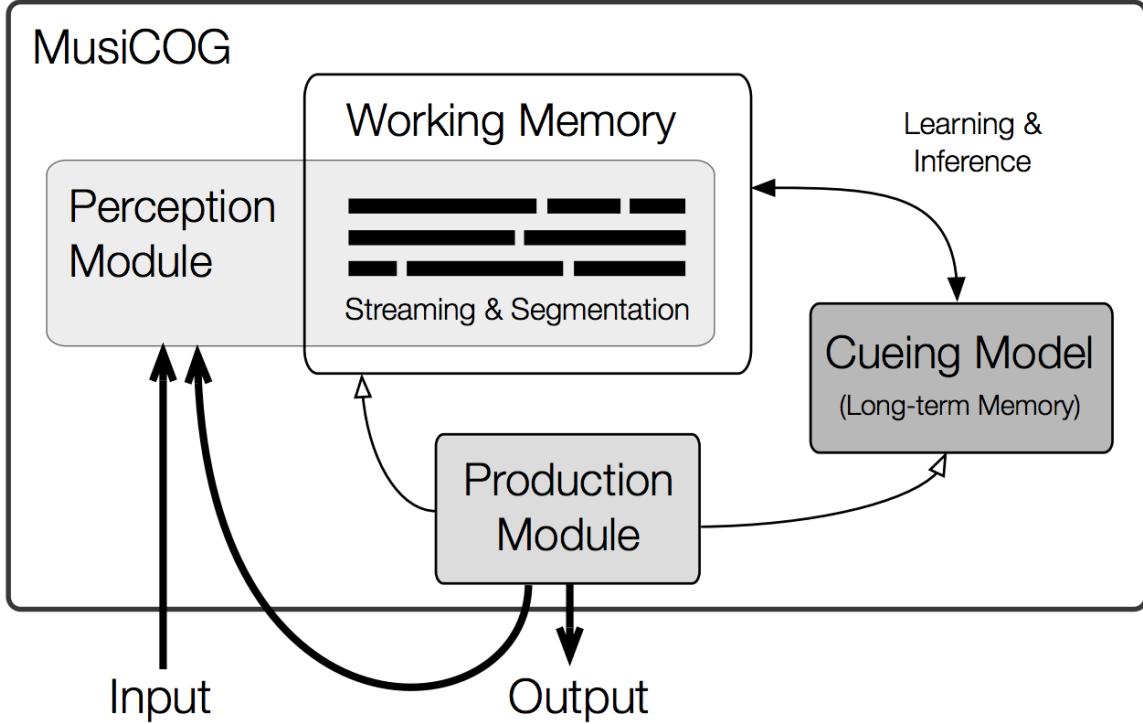


Figure 2.5: The architecture of MusiCOG [128]

audio corpus to use at the beginning of performance. The musical agents in this system generate sounds with concatenative synthesis using CataRT.

The third system with the BDI architecture is *MUSIC-MAS* ⑥ [144, 145]. The focus of this study was assisted composition and style imitation. *MUSIC-MAS* generates harmony progressions using organization-oriented MAS design to introduce flexibility and scalability to the system design. *MUSIC-MAS* implements a client-server based MAS architecture proposed by Ferber et al. [83] including multiple agent types. *ProviderAgent* assigns certain roles to *ClientAgents*. *MUSIC-MAS* has five client agent roles: *composer*, *evaluator*, *interface*, *data supplier*, and *control*. Composer agents implement BDI architecture to generate harmony progressions. Evaluator agents score the generated progressions using a fitness function that includes harmony progression constraints. These constraints are style specific and the authors state that the system is flexible enough to imitate any style by changing the set of constraints in the fitness function. The interface agents handle the interaction with the user. The data supplier agents collect and store all the data generated by the system. The control agents are responsible for the communication and coordination of the agents.

2.4.3 Cognitive musical agents with cognitive models

Maxwell et al. [128] propose the framework called *Hierarchical Sequential Memory of Music (HSMM)* ⑦. *HSMM* is build upon the *Hierarchical Temporal Memory (HTM)* [90]. The MuMe applications of HTM are recognition, generation, and continuation. Following *HTM*, Maxwell et al. [129]

present *MusiCOG* ⑧, a cognitive musical agent generating monophonic melody. *MusiCOG* is a mono agent system that applies the MuMe tasks of continuation and assisted composition. *MusiCOG* ⑧ listens to and learns from a sequence of MIDI data. *MusiCOG* consists of four modules: *perception module*, *working memory*, *cueing model*, and *production module* (Figure 2.5). The perception module implements segmentation as well as the generation of monophonic MIDI streams from a polyphonic input. Working memory is the short term memory that implements grouping of similar patterns. The cueing model is *MusiCOG*'s long term memory which learns the hierarchical structures between patterns in the working memory. Finally, the production module generates monophonic MIDI output using the knowledge representation in *MusiCOG*.

In regards to communication between musical agents, Murray-Rust and Smaill [143, 142, 141, 140] proposed the Musical Acts theory ⑨ that was inspired by the Speech Acts theory [107]. Murray-Rust and Smaill propose three qualities of Musical Acts:

- *Embodiment* through the production of music,...musical acts must have a manifestation in music.
- *Intention* is what differentiates a musical act from general musical playing. A musical act should have perlocutionary force.
- *Intelligibility* is necessary for a successful act; if it is not understood, then it will fail to change the world, as other musicians will fail to react to it.

The information in Musical Acts is generated by *descriptors*. A descriptor is a mapping from a *facet* (an aspect of music, such as melodic contour, chord, time signature) to a *value*. An *analyser* generates a descriptor in Musical Acts Theory. The theory also includes a set of performative actions (*inform*, *confirm*, *disconfirm*, *extend*, and *alter*) to create a dialogue between musical agents. Murray-Rust and Smaill also present a MAS with Musical Acts using a symbolic representation of music. The agents in the MAS were trained on the piece Canto Ostinato by Simeon ten Holt. The agents analysed levels, slopes, and patterns for note timing, length and loudness features using the symbolic music.

2.5 Reactive Musical Agents

Reactive agents respond to the changes in the environment without the explicit symbolic reasoning of the type carried out by cognitive agents. In MAS, there are two types of reactive agents architectures: *reflex* and *reactive*. Reflex agents do not have any internal states. The perceived states of the environment, *percepts* cause actions of reflex agents. Hence, the simplest agent architecture is a reflex system, that is a function that maps percepts to actions:

$$f : P \rightarrow A$$

where f is a function, P is a set of percepts, and A is a set of actions. Unlike reflex agents, reactive agents have internal states. These internal states are functional as opposed to cognitive.

A well-known reactive agent architecture is the Subsumption architecture [40]. The Subsumption architecture is hierarchical with multiple layers. Higher layers have a higher priority and vice versa. Outputs of higher layers can restrain, alter, or block outputs of lower layers.

Moreover, Brooks [41] discusses four key terms of the Artificial Intelligence and MAS research: *Situatedness*, *Embodiment*, *Intelligence*, and *Emergence*. Situatedness proposes that the intelligence is situated in the interaction with the environment, responding to percepts in a timely fashion, rather than reasoning about the environment through a symbolic representation. Embodiment is the idea that an agent is an ‘embodied intelligent agent’ and intelligence is situated in the real world through physical grounding. Brooks [41] claims that the interaction between an agent and its environment is the determinant of the intelligence of an agent. Therefore, intelligence emerges as a result of the interaction between the behavioral rules of the agent and its environment.

In the following, we survey 39 systems with reactive musical agents. We group these systems according to their environment types, reactive musical agents in real-world and virtual environments. Reactive musical agents in real-world environments appear in two categories: rule-based reactive musical agents and agents with Evolutionary Computation (EC). In the reactive musical agents in virtual environments, musical multi-agent systems simulate virtual environments to conduct a musical task. We group the reactive musical agents in virtual environments into two categories of multi-agent simulations with EC and multi-agent simulations with ecosystemic approaches. We differentiate the reactive musical agents that use EC to generate musical material from the musical MAS that use EC to evolve agents. The systems that we cover in Section 2.5.1 implement EC to generate musical material within a reactive agent architecture in real-world environments. We cover Multi-agent simulations that utilize EC to evolve agents in virtual environments in Section 2.5.2.

2.5.1 Reactive Musical Agents in Real-World Environments

We group musical agents in this category in two: rule-based reactive musical agents, and reactive musical agents with Evolutionary Computation.

Rule-based reactive musical agents

A recurrent theme in reactive musical agents is the application of music theory rules in the design of musical agents. Rule-based systems apply percept-to-action functions as *IF-THEN* conditionals. In most cases, these rules are strictly set by the designer of the system. We start our survey of rule-based reactive musical agents with the systems automatising the improvisation tasks.

One of the early musical agent is *Cypher* (10), a rule-based reactive musical agent working with symbolic representation of music [164]. Cypher is a multi-role, heterogeneous Multi-agent system. Cypher is also an example of *holonic* Multi-agent systems. In holonic systems, an agent is made of other agents [133]. In Cypher, the agents are hierarchically arranged and connected. At the highest level, there are two types of agents, listener and player agents. First, listener agents behave similar to the perception modules, analyzing the input data and providing high-level musical information

to player agents. The listener agent is made of the register agent, the dynamic agent, the density agent, the speed agent, the duration agent, and the harmony agent. Each of these agents implement a particular Music Information Retrieval (MIR) task. Second, player agents generate musical output in three ways: transformation of the input data (symbolic representation of music), triggering events by using the information provided by the listener agent, and choosing from a corpus of musical events. Rowe also mentions that the listener agents can be used as critics that analyze the output generated by player agents. Listener agents as critics report the success of previous events to player agents. This application of critic agents is similar to the reinforcement learning in statistical sequence modelling algorithms (see Section 2.6.1).

Voyager (11) is one of the well known musical agents [116]. Lewis mentions that the first version of Voyager goes back to 1986. The global behaviour mode of Voyager determines timbre, volume range, microtonal transposition, tempo, tactus (underlying, inner pulse), note probability distributions, pitch interval range, inter-onset time intervals, pitch set and pitch generation algorithm. Voyager has 64 MIDI-controlled, monophonic *player* agents. Depending on the global behaviour mode, Voyager groups or separates these 64 agents. Voyager changes the global behaviour mode every 5 to 7 seconds. The system includes 15 pitch generation algorithms and 150 microtonal pitch sets. Moreover, *setresponse* module handles the responses to the input data by modifying the parameters set by the *setphrasebehaviour* module. As of 2017, Voyager still performs in various venues. As a part of the Musical Metacreation concert at ISEA2015 in Vancouver, BC, Canada, Voyager improvised with two human performers playing prepared piano and clarinet.

Band-out-of-a-Box, Bob (12) is another one of the early reactive musical agents. Bob improvises with a human performer in the context of jazz and blues [180, 181, 182]. Bob focuses on the generation of melodies, specifically *solo trading* in jazz improvisation. In solo trading, musicians improvise one by one for a number of measures. The number of measures is an integer division of the total number of measures in a chorus. In the jazz context, it is common to trade solos in four or eight measures. Bob improvises with a human performer, along with a fixed musical accompaniment (Figure 2.6). Bob utilizes trees to represent a measure, and implements histograms of the pitch class, intervals, and melody direction to learn the playing modes of the human performer. The real-time generation of musical sequences is a mapping that associates the histogram of feature vectors to the playing modes of Bob. Bob generates these playing modes during the learning, and stores them as a knowledge representation. However, these playing modes do not provide a temporal pitch sequence. To generate a melody, Bob implements a first-order Markov chain in which the states consist of an absolute pitch value and a set of histograms. Thom also demonstrates Bob's performance by training two different agents on Stéphane Grappelli and Charlie Parker solos.

Adaptive Real-time Hierarchical Self-monitoring (ARHS) (13) emphasizes timbre in the reactive musical agents [104]. The system includes three modules that are *Sensing*, *Synthesis*, and *Processing*. The three module architecture of AHRS resembles Blackwell et al.'s [26] *PQf* musical agent framework in which P is the machine listening and analysis; Q is the synthesis, and f implements reasoning and generative functions. We further discuss the *PQf* approach in Section 2.8.4. The

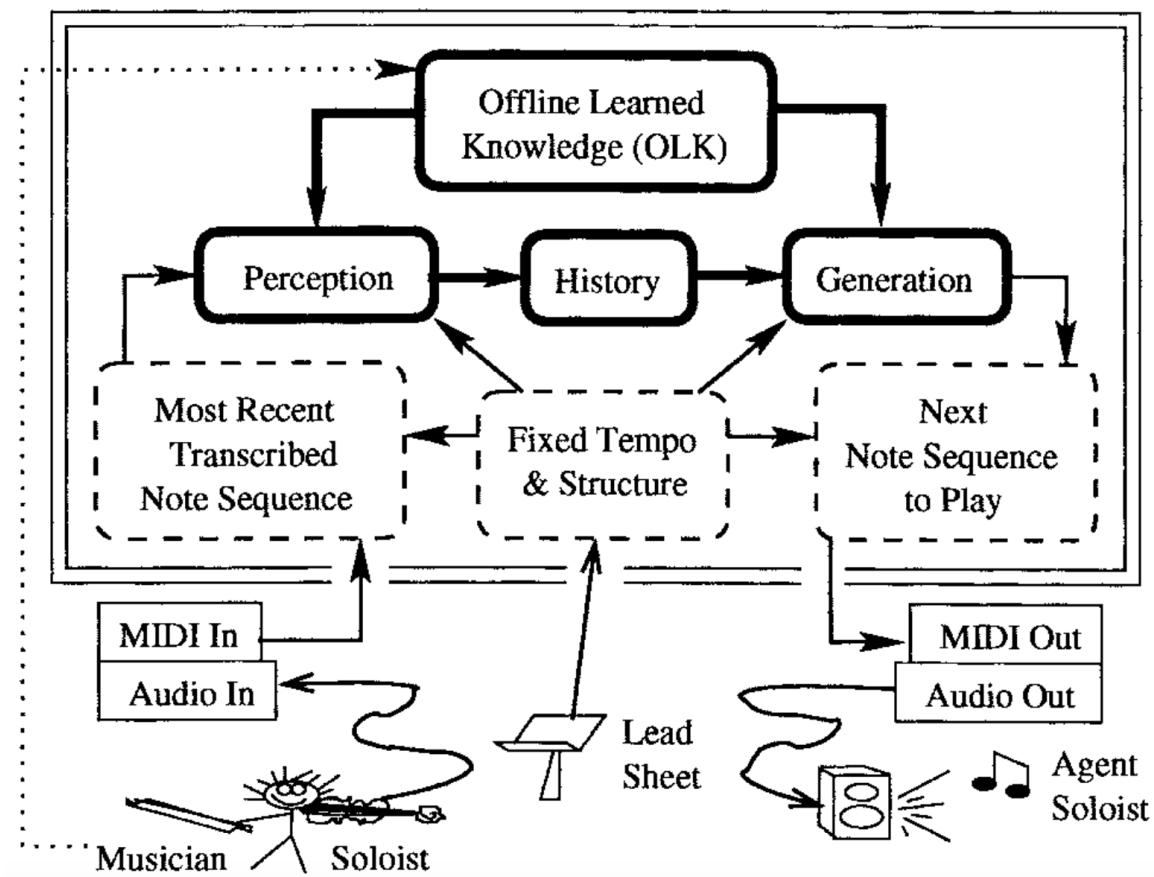


Figure 2.6: Bob's system architecture [182]

sensing module of AHRS includes pitch, amplitude, loudness, tempo, auditory roughness, and timbre analysis. The sensing module outputs a performance mode that is updated every second. Then, the processing module uses the performance mode to choose sounds. There are two features in the processing module: short-term responsiveness and long-term adaptivity. The system initiates short-term responses using the Sensing module output and fuzzy logic. Short-term responses are beginning a phrase, ending a phrase, and changing the timbre. The long-term adaptivity implements fuzzy sound selection based on 8-second windows of the Sensing module output. The Synthesis module includes one or more agents. The Sensing and Processing modules set global variables for these agents. Agents generate parameter curves to change pitch and timbre characteristics.

ListeningLearning (LL) (14) is a system that performs with humans in the context of free improvisation [50]. The listening algorithm has three modules: rhythm tracking, silence detection, and timbral state clustering. The rhythm tracking module calculates onset detection, periodicity, and inter onset interval. Also, the rhythm tracking module differentiates free time playing from steady metric tempo. The silence detection module utilizes perceptual loudness. LL clusters timbral states using the audio feature statistics of cosine basis energy, cosine-wise inter-frame flux, RMS amplitude, spectral centroid, spectral irregularity, and spectral energy. The author normalizes six features of timbral state clustering using the *adaptive distribution model*. The adaptive distribution model uses the statistics of feature vectors to implement a normalization that is similar to histogram equalization in Computer Vision. LL implements the timbral state clustering using online k-means clustering with the Euclidian distance metric to follow the timbral choices of the human performer. LL includes ten agents to generate output. Each agent correspond to one of ten timbral states. Only one agent is activated at a time depending on the timbral state calculated by the machine listening module. Each agent has a unique set of parameters for audio synthesis and processing modules. Each agent has four audio synthesis and processing modules: sample based drum kits, a physical modelling synthesis of the vocal tract, a four-voice subtractive synthesis, and fifty different audio effects. Collins states that using ten agents provides the system the diversity of responses required for free improvisation with human performers.

Virtualband (15) is a MAS in which musical agents imitate playing styles of musicians [137]. An agent learns from recordings of a musician to imitate the style of the musician. Each agent is also provided with a lead sheet including the chord progression during learning. Hence, agents are also aware of the harmonic content. Two types of agents appear in Virtualband: *master* and *slave*. Slave agents follow the master agent's eventfulness. The master agent can be a human performer or a software agent. The slave agents have a hybrid corpus including audio excerpts and the audio feature distributions of these excerpts. The duration of excerpts are set to one beat or one bar before the learning. In the generation mode of Virtualband, a slave agent generates output by using concatenative synthesis and chooses audio files to play by following the eventfulness of the master agent. Slave agents conduct the follow role using two mappings: the mapping between audio feature distribution of the master agent and the audio feature distribution of a slave agent, that is, *feature-to-feature (f2f)* mapping. *f2f* uses percentile function to map between two feature distributions. The

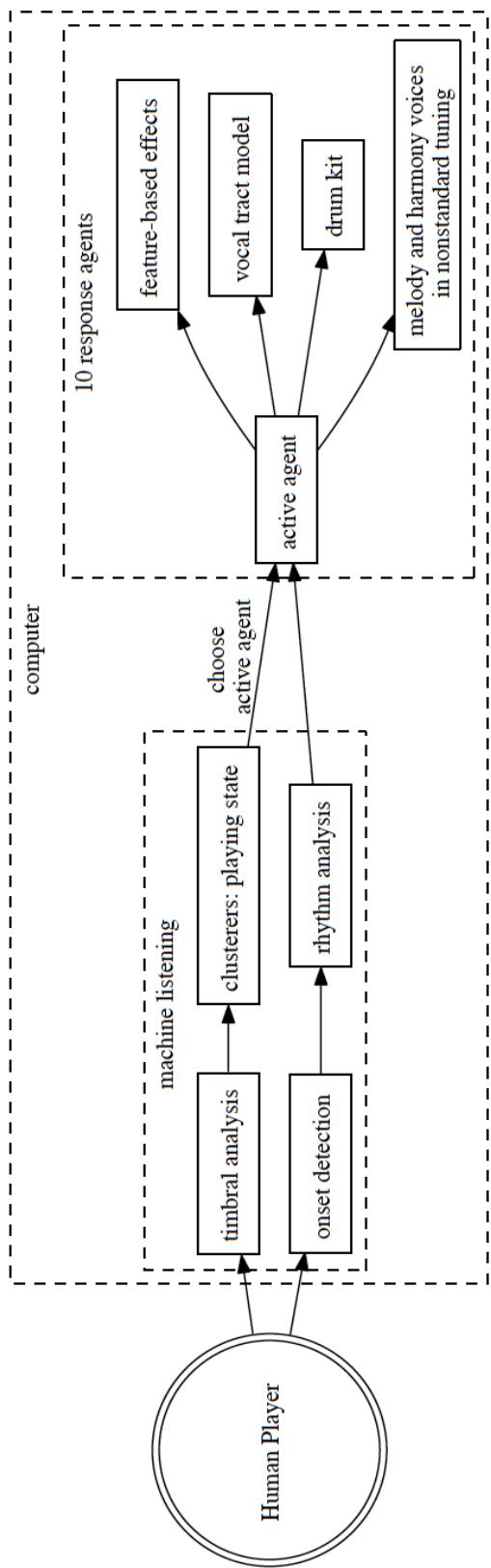


Figure 2.7: The architecture of LL [50]

authors also present case examples of jazz improvisation, interactive mash-ups, and beatboxing with Virtualband. The advantage of Virtualband is that the symbolic corpus is not necessarily tied with the audio corpus. That is, we can use the same feature distributions with different audio corpora. Examining the example performances of Virtualband, we observe that the algorithm is successful on jazz improvisation. However, it is not clear why the authors choose the percentile based *f2f* mapping. In the examples of Virtualband, agents are trained on a small corpus, performances of one or two songs.

Odessa (16) is a reactive musical agent based on the Subsumption architecture [117]. Odessa is a mono-agent system that plays improvised music using audio input and MIDI output. Arranged from the lowest to the highest layer, there are three layers in the architecture: *play*, *adapt*, and *diverge*. The *play* layer generates the MIDI output. The *adapt* layer alters the output of the *play* layer according to the input when Odessa listens to other performers using machine listening algorithms for pitch and loudness estimation in the *adapt* layer. The *diverge* layer constructs higher level musical section changes.

Following the systems focusing on the improvisation tasks, we present five systems generating rhythm: (17), *VirtuaLatin* (15), *DrumTrack* (19), *BBCut2* (20), *Kinectic Engine* (21), and *BeatBender* (22). Pachet [151] presents a rule-based reactive agent (17) that generates rhythm and harmony progressions. The agents implement two sets of rules to produce rhythms. The first set of rules create rhythms from scratch using three rules: *emphasize strong beat*, *emphasize weak beat*, and *syncopation*. The second set of rules generate variations of existing rhythms with three rules: *add random*, *remove random*, and *move pitch*. To generate harmony, the agent applies the same rules in the vertical dimension (pitch) as opposed to the horizontal dimension (time). Also, the agents apply two additional rules (called *attraction* and *repulsion*) in the horizontal dimension (time) to handle harmony progressions.

Murray-Rust et al. [139] present another rule-based, rhythm generating MAS. *VirtuaLatin* (18) generates *timbales* accompaniment to Salsa music that is pre-recorded as MIDI files. *VirtuaLatin* works offline. The perception module extracts higher level information from input data on four dimensions: activity, harmonic information, rhythmic information, and musical section. The agents select one rhythm from a corpus using the symbolic representation of the song. The agents apply three more rules to introduce ornamentation: *phrasing*, *chatter*, and *fills*.

DrumTrack (19) is another musical agent generating rhythm to accompany a human improvisor [47]. DrumTrack focuses more on how to track the tempo of a human performer real-time using machine listening. Although the architecture of DrumTrack is reported as rule-based, the details are not presented.

BBCut2 (20) is the second version of Collins' [48] musical agent generating *breakbeat*. Within our knowledge, the details of the first version is not published. Breakbeat refers to either a musical

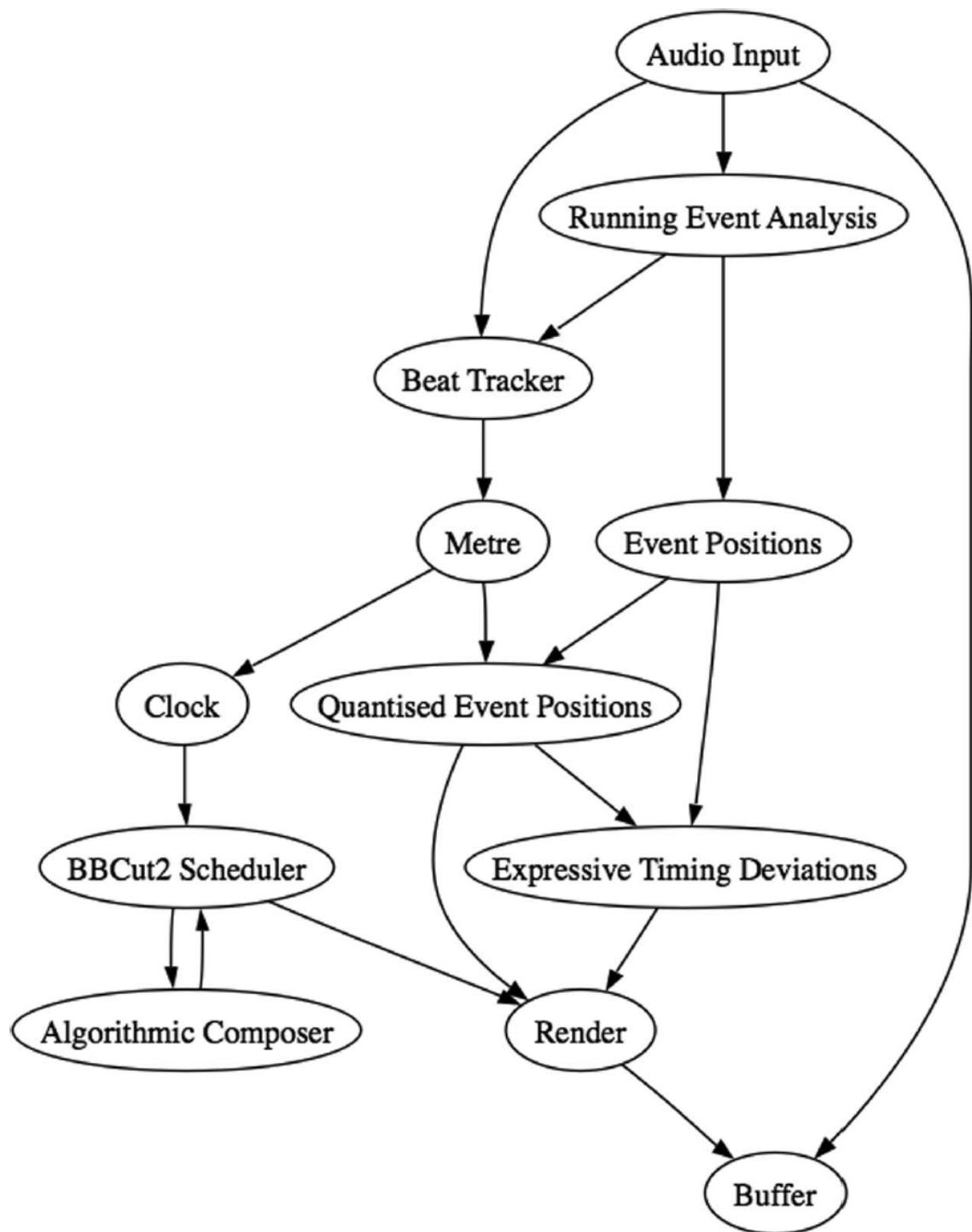


Figure 2.8: The structure of BBCut2 [48]

section where all instruments stop and a drum solo begins⁷, or the electronic music genre in which breakbeat drums are sampled to make mash-ups. BBCut2 uses machine listening techniques to analyze an audio signal and implement beat tracking and segmentation (Figure 2.8). The agent has an algorithmic composer module. The rules in the algorithmic composer module include generative streams of events, static sequences, shuffling the order sequences, weighted choices, nested patterns, and more.

Eigenfeldt [67] presents the *Kinectic Engine* (21), a rhythm generator with multiple reactive musical agents. The Kinectic Engine includes two types of agents: *conductor* and *player*. There is only one conductor agent, and this agent handles user interaction, identification of player agents, communication between player agents, and sending the tempo to the player agents. Eigenfeldt implements *personality traits* to vary the player agents' behaviors. Personality traits are presets that determine the density, the amount of rhythmic variation, and timbre of an agent. Player agents decide to play at a certain time using fuzzy logic. The author proposes two types of rules to generate rhythms with player agents: density spread and pattern matching. Moreover, the communication between player agents has two interaction modes: rhythmic polyphony and rhythmic heterophony. The player agent decides on the interaction mode by calculating the rhythmic similarity rating between the agent's own output and the other agents' output. Higher similarity results in the alteration of generated output to satisfy rhythmic heterophony. Likewise, lower similarity alters the agents' generated output towards rhythmic polyphony. The author develops the Kinectic Engine further by introducing Evolutionary Computation to the system (see Section 2.5.1).

BeatBender models a drum circle with the Subsumption architecture to create emergent musical rhythms [114]. Each rule has an antecedent and a consequent. Antecedents dictate a set of preconditions for a specific rule to be activated. The consequent is the result of an agent's actions when a rule is applied. Each beat, agents decide to play (or not) using their percepts and a set of rules. *BeatBender* includes four types of rules:

- *Collective*: rules that use the total number of active agents
- *Directed*: rules that check states of specific neighboring agents
- *Temporal*: rules that use the histogram of an agent's states
- *Undirected*: rules that check states of any neighboring agents.

Following the musical agents applying musical tasks of improvisation and rhythm generation, we continue our survey with the remaining rule-based reactive musical agents. *Andante* (23) is a musical agent framework for mobile platforms [187]. The framework is inspired by the client/server model in MAS. The implementation rests on Aglets Software Development Kit and JAVA Sound

⁷One of the most famous breakbeat sections is the *Amen Break*, which is a 4 bar drum break of the song "Amen, Brother" by 1960s soul/funk band the Winstons. The recording of Amen Break can be found at http://en.wikipedia.org/wiki/Amen_break.

API. The authors also present a MAS with agents generating monophonic melodies using different types of noise such as pink, white, and brown noises. Although the generation is MIDI, there are built in synthesizers that agents can choose from, to generate the audio output.

Public Space Interactive Web-based Composition System (*PIWeCS* (24)) is a browser-based multi-agent system that generates musical compositions [197]. The system has an audio corpus of Maori instrument samples. The user can set three variables: *unity/variety*, *volume*, and *tempo*. The interface includes a conversational model of interaction between the machine and the user. There are four agent types in PIWeCS: *reception*, *helper*, *learner*, and *extender*. However, the details of the system architecture as well as the user interface and interaction model are not disclosed.

Eigenfeldt and Pasquier [75] present a unique application of generative soundscape composition with MAS including four musical agents, called *Coming Together:Freesound* (25). The system is a part of the musical agent series called *Coming Together*. The agents react on the environment, and also communicate using the blackboard architecture to choose sounds to play. The corpus of an agent is labeled by spectral contents and the metadata tags of *voices*, *animals*, *water*, and *outside*. There are three types of agent interactions in this system: sharing the metadata tags on a shared blackboard, reacting to the spectral content of the sonic environment, or both.

The fourth version of the *Coming Together* series includes a musical agent (25) that concentrates on musical arrangement [77]. The other agents in the system generate a corpus of musical sections with symbolic music representation (as MIDI files). The arranger agent chooses a random musical section from the corpus, that is, a MIDI file as the first movement. Then, the arranger agent calculates the similarity of the first musical section and the remaining ones on the dimensions of *cumulative density*, *pitch range and variation*, *volume*, *overall length*, *specific instrument presence*, and *harmonic movement*. The arranger agent sorts the remaining musical sections and chooses the next musical section using Gaussian selection. The agent repeats this process using the previously selected musical section. The agent stops when a user-defined duration has reached.

ParamBOT (27) is a curator agent that generates musical sections of Moment form [72]. *ParamBOT* initiates or stops other musical agents to create distinct musical sections. This implementation explores Stockhausen's idea of Moment Form in experimental music. A musical section in Moment form is free of the previous and next sections. At the beginning of each musical section, *ParamBOT* sets global parameters of *speed*, *activityLevel*, *voiceDensity*, *complexity*, *volume*, *consistency*, and *pitch*; and initiates other musical agents within the Musebot framework. We introduce the Musebot framework in Section 2.8.

Reactive musical agents with Evolutionary Computation

Evolutionary Computation (EC) is an abstraction of Darwinian evolution. EC implements *survival of the fittest* to find a solution to a given problem using a population of solutions. EC applies the genotype-phenotype dichotomy. A genotype is a representation of a solution (*phenotype*). A *fitness function* evaluates phenotypes (solutions) by assigning *fitness scores*. Genetic operators, *crossover*, *mutation*, and *reproduction*, generate new solutions called *offsprings*. These offsprings go through

ParamBOT

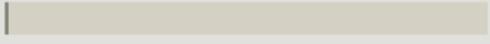
06.18.16

arne eigenfeldt

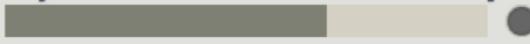
duration (sec): 600

play

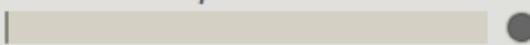
section progress:



dynamic Parameter Probability:



relative Proportions:



generate

elapsed time: **00:00**

section: **0**

of sections: **7**

5 shortest sections: (secs): **14. 30.
66. 89.
90.**

speed



pitch

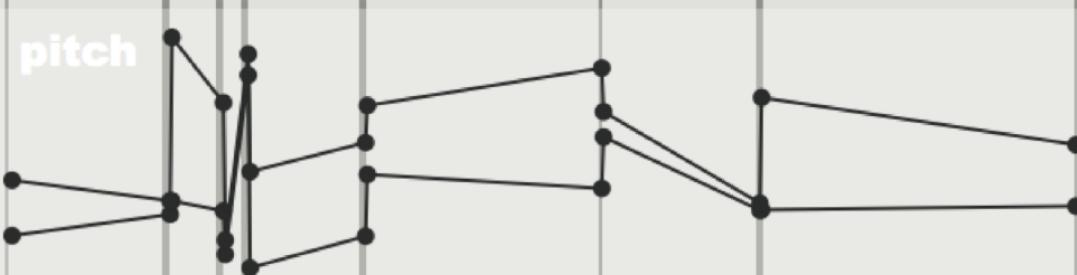


Figure 2.9: *ParamBOT*, a curator agent implemented using the Musebot framework in MAX [72]

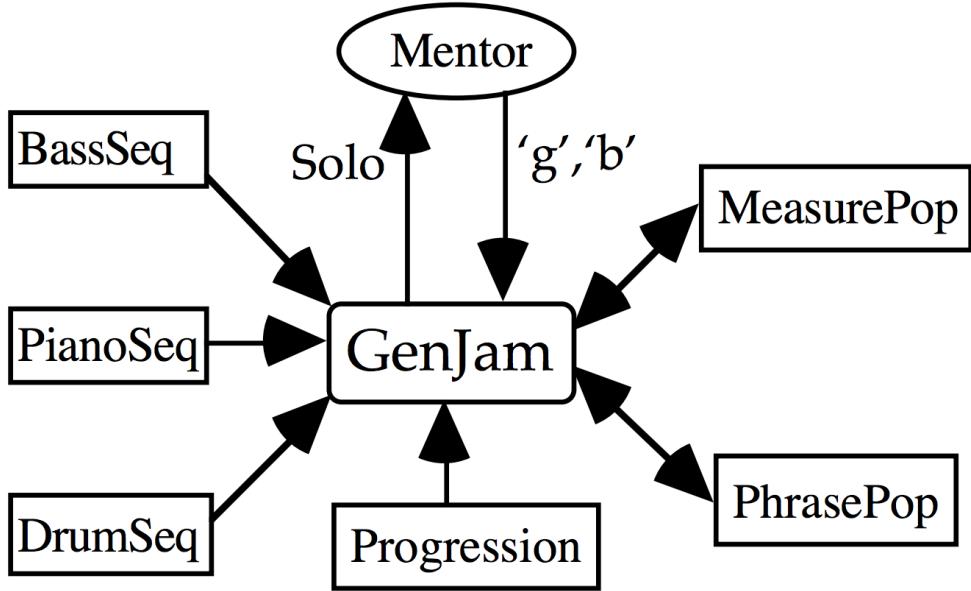


Figure 2.10: The block diagram of GenJam [22]

a selection process to create next generations. An EC algorithm continues evolving a population until *stopping criteria* are reached [169]. EC has been widely used in optimization problems such as synthesizer preset generation [178]. Although the problems in MuMe research do not necessarily involve optimization, EC has also been widely used in the MuMe context [135]. In this section, we cover musical agents in real-world environments that use EC to generate musical material for the agent.

We begin by presenting three systems applying improvisation tasks. *GenJam* (28) is one of the early musical agent systems [22]. *GenJam* implements a sub-branch of EC algorithms, called Genetic Algorithms (GAs) to generate melodies. *GenJam* improvises Jazz Music using an Interactive Genetic Algorithm (IGA) that generates musical phrases using a symbolic representation of music (MIDI). In IGAs, the user evaluates and scores all individuals in the GA population every generation. *GenJam* concentrates on jazz improvisation over a given chord sequence. Learning, breeding, and demo are the three modes of *GenJam*. In the learning mode, a human listener gives real time fitness scores to *GenJam*'s musical phrases. The listener labels the *GenJam*'s current solo with the labels 'g' and 'b' that stand for 'good' and 'bad' respectively (Figure 2.10). In the breeding mode, *GenJam* implements genetic operators on the population of musical phrases. *GenJam* replaces half of the population every generation with new offspring using crossover and mutation. In the demo mode, *GenJam* improvises on a Jazz tune, using a chord progression file. Although Biles does not present *GenJam* as a musical agent, we can analyze the system architecture as a musical agent, with inputs of chord progression, rhythm section, and human evaluation of the generated phrases, and with an output of solo jazz improvisation. Biles, who is also a trumpet player, presented *GenJam* in numerous concert venues as GenJam being another jazz player [23].

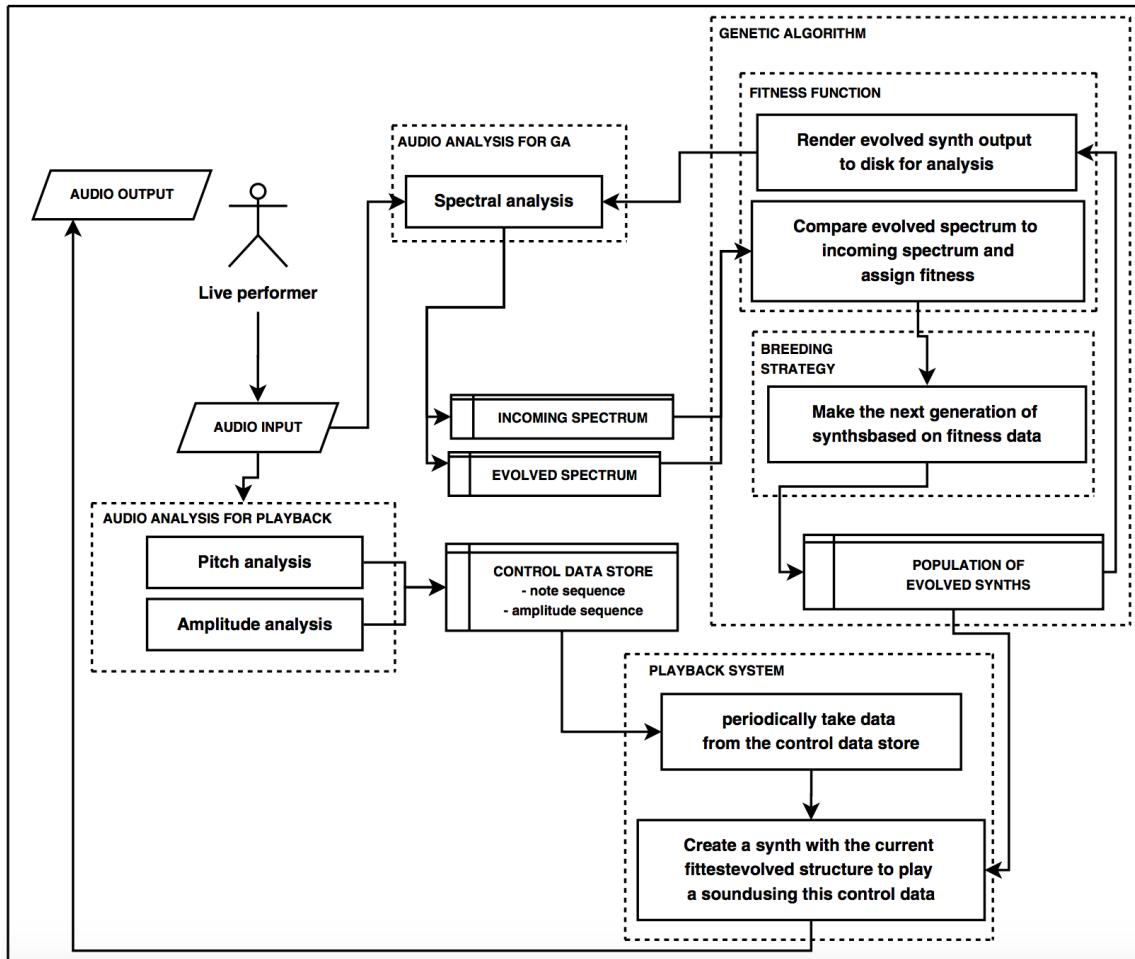


Figure 2.11: The system architecture of Yee-King's [204] musical agent

The following two systems that apply improvisation tasks imitate the style of human performers using EC: the system (29) and *Frank* (30). Yee-King's [204] study comes forward with a unique implementation of evolving timbres, rather than evolving symbolic representations of music. Yee-King presents a reactive musical agent (29) for live performances with human performers. This study also implements IGA with the Java programming language and embeds SuperCollider's *scsynth* framework for sound synthesis. The agent records the pitch and amplitude information of the live input to its memory to imitate the improvisation of other performers. The implementation includes two audio synthesis techniques: Additive Synthesis and Frequency Modulation (FM) synthesis. A control data of pitch and amplitude provided by the memory of the agent manipulates the parameters of these synthesizers. The IGA optimizes the timbre of the audio synthesis to match the agent's timbre to the other performers. The author presents the initial results of this system, without recombination in the genetic algorithm. Hence, breeding is mutation only in the initial experiments. The author also comments on the improvisational skills of the agent, saying that the agent has a 'responsive feel' with 'unique dynamic sound.'

Frank (30) is a musical agent that evolves *lexemes* to imitate a human performer [158]. Lexemes are clusters of MPEG7 vectors ⁸ The clustering method of Frank is k-NN and the cluster locations are the centroid vectors. EC implementation includes two genders: *male* and *female*. *Frank* introduces the input frames as new female individuals to the population. During reproduction, the offspring gender is set to either male or female randomly. The authors clarifies that female agents function as critic agents and the implementation of two genders introduces criticism to the system. The fitness function uses Euclidian distance between the input frame vector and an individual with an application of *imprecise pattern matching* with weight matrices. To implement imprecise pattern matching, the authors utilize a similarity threshold that female individuals use to choose a male individual to mate. The authors state that this balances coherence and novelty in the system. *Frank* plays back winner frames from the population.

The next two applications implement MAS with EC to generate rhythms. Gimenes et al. [95] implement Dawkin's idea of memes in reactive musical agents. The system is called RGeme (31) and the MuMe application of interest is rhythm generation with symbolic representation. RGeme includes three types agent tasks: *listening*, *practicing*, and *composing*. The listening and practicing phases constitute the learning whereas composing is the generation. Each agent also has an evaluation algorithm to choose which music files are used in the learning tasks. During the learning, agents generate a *Style Matrix* in which the rhythmic memes are stored. The weights of rhythmic memes are determined by the number of times it is encountered and in which listening cycles it is encountered. To introduce temporality to the agent memory, rhythmic memes also lose weight if they are not encountered in later listening cycles. The authors also present analysis of case studies in which agents are trained with Brazilian music composed by Chiquinha Gonzaga, Ernesto Nazareth, Jacob do Bandolim, and Tom Jobim. The generation phase was not implemented in this version. In

⁸MPEG 7 is a standardization of low-level feature calculation and thumbnailling for multimedia.

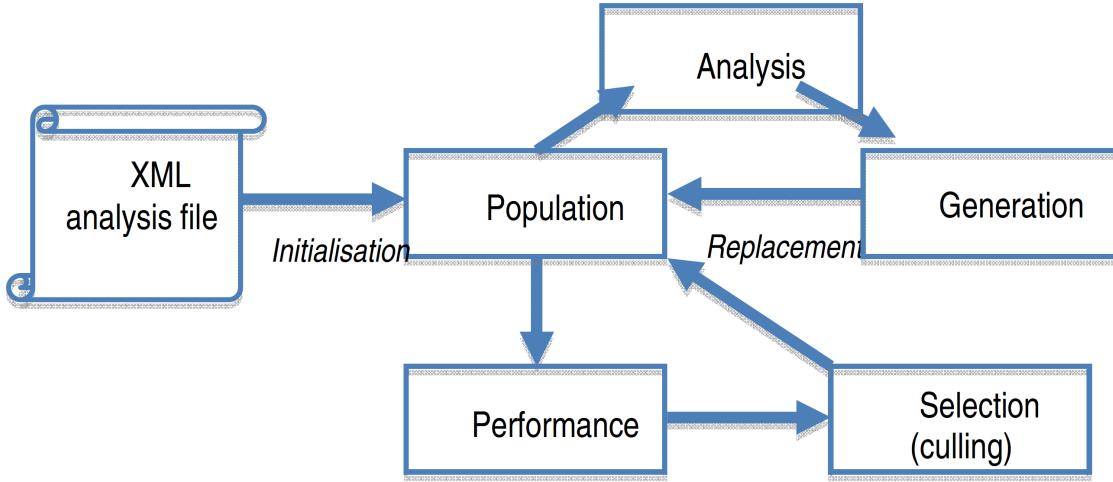


Figure 2.12: The block diagram of Kinectic Engine version 3 [68]

the following years, the generation phase was also implemented and this system is used as the brain for a robotics implementation [96]. However, we exclude robotics implementations in this review.

The latest version of *Kinectic Engine* (21) also applies rhythm generation with EC [68, 70]. This study aims to generate rhythms that continuously evolve, rather than rhythms emerging, or appearing. Eigenfeldt states that EC provides a musical memory; and thus, introduces temporality. The system analyzes a corpus of rhythms (MIDI files) offline. Moreover, *Kinetic Engine* analyzes the individuals in the population on two musical dimensions: density (the number of events) and complexity (the degree of syncopation). EC in *Kinectic Engine* implements Roulette Wheel Selection, a well-known selection algorithm in EC [169]. The author mentions that *Kinectic Engine* includes a crossover-like breeding using a single parent; however, the details of this crossover-like breeding are not disclosed. The system also implements mutation. The population of rhythms are provided to all player agents. A player agent chooses individuals (rhythm patterns) in the population using user-set global density and complexity parameters. The player agent utilizes a k-nearest algorithm to find rhythm patterns with the user-given density and complexity values in the population. Eigenfeldt and Pasquier [74] present artistic implementations of the system along with the artistic evaluation of Kinectic Engine, concluding that following versions of the system should introduce ‘intelligent’ melody and harmony generation.

Aucourtier [11] implements a multi-agent society (32) to evolve tuning systems. The fundamental frequency and the timbre are the global variables of the system. The agents include a dissonance calculation formula that is the parameterization of the experimental Plomp-Levelt curves ⁹. Each agent has a tuning system with the same number of notes. One agent listens (*tuner role*) while the other plays (*player role*). The tuner agent tunes its notes by minimizing the dissonance between the

⁹Given a root note, Plomp-Levelt curves were proposed to calculate consonance or dissonance of any other note to the root note [159].

player agent's note and its scale. There are two types of interactions between agents: *single note shift* and *drone shift*. In single note shift, the player agent chooses a random note to play and the tuner agent tunes one note that is chosen randomly. In drone shift, the player agent plays one note and the tuner agent tunes all notes in its memory by minimizing the dissonance. The environment includes two types of timbres as global variables: harmonic timbre (where the partials are the integer multiplication of the fundamental frequency) and compressed timbre (where the partials are spaced narrowly as stated by a geometric law).

2.5.2 Reactive Musical Agents in Virtual Environments

A recurrent theme in musical agent studies is the applications of self-organizing agents that situate in virtual environments. The authors define the dimensions and properties of virtual environments. Systems generate music using the spatial orientation of agents and/or virtual encounters between agents. This data is mapped to parameters of audio synthesis, or symbolic representation of music. Hence, the complex behaviors in a virtual environment create music.

Bown et al. [31] propose five elements of ecosystemic creative domains: *space*, *materials*, *features*, *actions*, and *processes*. In ecosystemic creative domains, an agent situates in *space*, perceives the environment using *features*, performs *actions* using *materials*, and the environment changes due to *processes*. In the following sections, we categorize musical agents in Virtual Ecosystems in two groups.

Multi-agent Simulations with Evolutionary Computation

In this section, we cover musical MAS in which the system uses EC to evolve agents. Following, we present two implementations of melody generation. Todd and Werner [185] present a system (33) that evolves monophonic melodies through agent interactions. The implementation is inspired by mating calls and singing rituals of wild animals in nature. There are two types of agents, referred by the authors as *male/composer* and *female/critic*. Male agents have a 32 note melody that spans two octaves. Each female agent includes a Markov model transition matrix indicating the probabilities of note transitions to rate the melodies of male agents. Each female listens to a subset of randomly selected male agents. After listening to all male agents in the subset, female agents choose one male agent to mate. The system uses crossover and mutation operators to generate offspring. The selection process of this system is not disclosed.

Similarly, *Living Melodies* (34) is a musical agent system that is also inspired by mating calls in nature [57]. There are two genotypes in Living Melodies: sound and procedural genotype. Sound genotype dictates how an agent listens to other agents and how an agent generates sound. Procedural genotype designates how an agent interacts with and traverses in the ecosystem. The agents born when two agents mate, but there are no genders in the system. The system creates the genome of offspring using crossover and parents' genome. Moreover, each agent is born with an energy level. The agents loose energy points as they act within the environment. An agent dies when its energy

level is below a threshold or a global, preset maximum life span has been exceeded. There are different configurations of sound mapping in the system. The agents generate mating calls as MIDI outputs using the information coded in their genome and communicating with other agents. The authors reported that the system can generate recurring patterns.

Martins and Miranda [126] presented a system (35) that generates rhythmic phrases. This implementation focuses on the abstraction of music as a cultural phenomenon driven by social pressure (the system number 31 in Table 2.1). Although this study includes an A-life algorithm rather than EC, the ideas of survival of the fittest, breeding and assessment of a fitness score also appears in this implementation. The system includes a population of agents that are identical in the system architecture. The agents situate in a 2D space in which they interact with each other. Each agent has a memory of rhythmic phrases. During each interaction, one agent takes the role of *player* and the other takes the role of *listener*. As a consequence of the interaction between two agents, each rhythmic phrase of the player agent is given a popularity score by the listener agent. If the listener agent recognizes a rhythm of the player agent, the listener agent gives a higher score to that rhythm, or vice versa. Moreover, the popularity of all rhythms drops by 0.05 after each interaction to introduce *aging* to the system. A transformation algorithm is applied to each rhythm that is shared between two agents during an interaction to foster novelty in the system. Martins and Miranda [127] further analyze the system measuring the similarity of rhythms. The analysis includes size and complexity of an agent's rhythm memory, the similarity, and clustering of agents, lifetime and novelty of generated rhythms. The authors state that the system exhibits 'the emergence of coherent repertoires across the agents in the society' in which the size of an agent's memory can be controlled using the popularity parameter.

Miranda et al. [134] present a system that evolves expressive performance of music using an imitative multi-agent system, called *Imitative Multi-agent Performer (IMAP)* (36). The authors define *expressive music performance* as the performance strategies that are not explained in the score, also known as the problem of *interpretation* in the context of MuMe. *IMAP* uses an *imitative model of behavior transmission*, that is similar to the GA model of behavior transmission. In both of these models, the algorithms generate a population of agents whose behaviors are defined by a genetic code. The difference between these two models is that the GA model uses a global fitness function whereas the imitative model has a non-global fitness function where each agent has a different fitness function. In the imitative model of behavior transmission, an agent shares its behavior to the other agents. Agents evaluate this behavior using their fitness function, and if the behavior scores high enough, the behavior of the evaluating agent is updated accordingly. Hence, an agent has two functions: performance and evaluation. The parameters of the interpretation task are tempo and the loudness deviations. The fitness function is rule-based, implementing five rules of *performance curves*, *note punctuation*, *loudness emphasis*, *accentuation*, and *boundary notes*. The rules have weights that are particularly set for each agent. These weights make agent's fitness function unique.

Another implementation that combines biologically inspired algorithms with MAS is *River-Wave* (37) [130]. The researchers explore the idea of niche construction in ecosystem modeling to

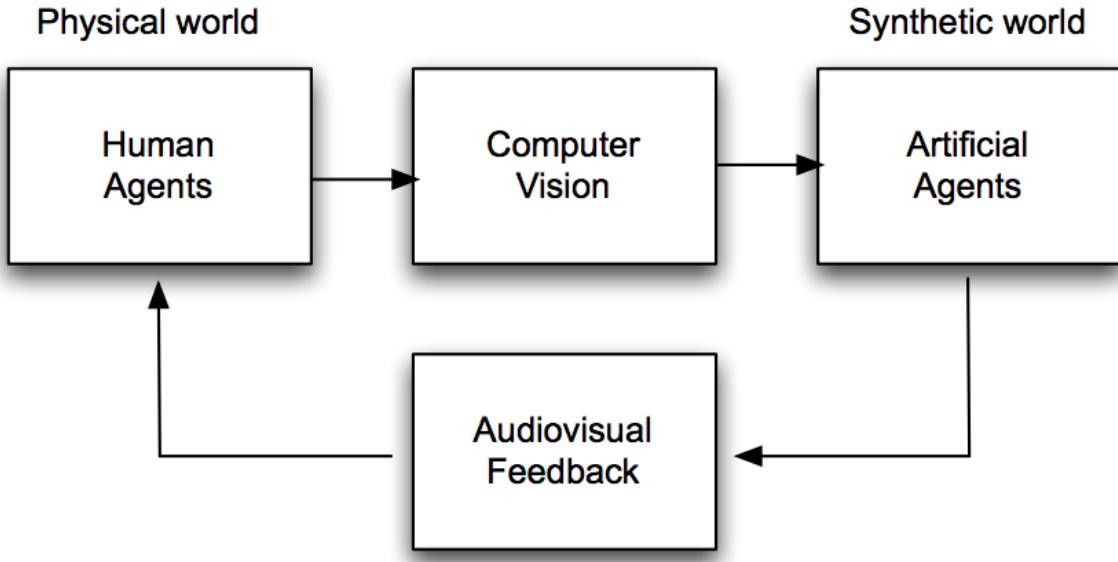


Figure 2.13: The system design of *Petri* [20]

create digital art and music. Niche construction is the phenomena of organisms establishing a more habitable environment for their offspring, which can also be approached as a process that precedes the evolution. The cooperation between agents becomes more prominent since parents aim for more habitable environments for their offspring. The musical agent system, *RiverWave* is a one dimensional, toroidal ecosystem that controls an additive synthesizer. Each agent location determines the frequency of the oscillator. Each agent has a height variable and agents affect the height of the neighboring agents. The height parameter of an agent is mapped to the amplitude of the oscillator.

Petri (38) is an interactive audio-visual system that utilizes a virtual environment and the real-world interactions [20]. The reactive agents situate in a virtual environment while the parameters of virtual environment change with a computer vision input (Figure 2.13). A webcam input is processed with a computer vision algorithm that provides five visual features to the virtual environment. These features define attraction points in the virtual environment. The agents move closer to these attraction points. There is also a life cycle that each agent goes through. New agents are created closer to the attraction points. The reactive agents have genders and communicate with each other. The communication between neighboring agents results in the sound generation. When agents decide to generate sound, the location of the agent defines the synthesis parameters. The 2D virtual environment is mapped to the FM synthesis parameters of carrier frequency and modulation frequency.

Multi-agent Simulations with Ecosystemic approaches

Blackwell and Young [25] present two applications of swarming in symbolic music generation. Swarming is a multi-agent behavior that is inspired by the behaviors of animal herds like birds, fishes, and insects. The behaviors emerge as a result of four rules: agents try to move closer to

neighboring agents, neighboring agents avoid collisions, all agents try to match velocity including the direction, all agents try to move towards attraction points. Likewise, self-organization has four components: positive feedback, negative feedback, amplification of fluctuations, and multiple interactions. *Swarm Music* (39) and *Swarm Granulator* (40) are ecosystemic reactive agents where the agents situate in a virtual environment. In *Swarm Music*, the authors propose a mapping between the spatial locations of agents and symbolic music parameters of pitch, loudness, inter-onset interval, duration, chord number, and sequence number. The authors also propose the idea of using two swarms for symbolic music generation where the spatial locations of one swarm are the attraction points of another. In *Swarm Granulator*, the agent records the human performer in an audio buffer while calculating the audio features of the pitch, amplitude, duration, and duration between successive sound-events. The swarming outputs six parameters audio buffer transposition, amplitude, duration, the time between successive grains, grain attack and decay time.

Ando and Iba [4] propose a musical agent system (41) including a virtual environment in the application of extended instrument design (the system number 36 in Table 2.1). Although the authors claim that the system is a cellular automata implementation that includes a MAS, the details of the system design is not disclosed. Ando and Iba say that the states of agents in the cellular automata change according to some pre-defined rules while a human performer plays a MIDI keyboard. The details of these rules are also not presented in the study.

McCormack et al. [131] propose a unique idea of using a 2D virtual environment with musical agents as a dynamic graphic score that generates music. The name of this framework is *Nodal* (42) and available as a commercial software ¹⁰. Users create a virtual environment using *nodes*, *edges*, *node traversals*, and *player agents*. Users put nodes in the environment and create connections between these nodes. These connections are called *edges*. Player agents traverse the nodes and edges. Each time a player agent reaches a node, the agent plays a MIDI note and changes its state variables (lists of pitch change, note-on, and note duration, MIDI instrument). The authors also give examples of bi-directional and asymmetric cycles, and cyclic pitch phasing as the examples of emergent behaviors in *Nodal*.

OSCAR (43) is a MAS with reactive agents situating in a virtual environment [16, 19]. This application focuses on the problem of generating non-idiomatic improvisation (a.k.a. free improvisation) with the symbolic representation of music. This study focuses on *autopoiesis*; that is ‘the continuous creation of new answers while facing an unpredictable environment. Agents situate in a 2D environment having parameters of *physical position*, *energy level*, *distance of communication*, *distance of neighborhood*, *activation*, *orientation*, *affinities*, and *personality dataset* of *pitch intervals*, *durations*, and *velocities*. The system tries to minimize overall social stress by using affinities between agents. The system generates musical output using the histogram of agent communications on each iteration. An agent that initiates a musical event generation chooses one of two methods: *contraction*, and *expansion*. Contraction generates a single musical event using a set of events, whereas

¹⁰<http://www.nodalmusic.com/>

expansion generates supplementary events using a single source event. The authors also presented three experiments with the systems to show emerging patterns. The system presented periodic patterns running over longer durations as well as complex behaviors.

Eigenfeldt and Pasquier [76] use Concatenative Synthesis with an ecosystemic MAS. The authors proposed the idea of generating music through consumption of virtual food in a virtual environment. The system is referred as *Shoals* (44) that is a part of series of generative music systems, called *Coming Together*. The system uses Concatenative Granular Synthesis with CataRT, an external library that is available in the visual programming language MAX. The real-time audio feature extraction of audio input creates food in the virtual environment that agents situate. The agents can move within the virtual environment. As an agent finds and consumes food, the consumption is sonified using CataRT. The agents are randomly initialized with synthesis parameters of *grain duration, delay between grains, amplitude, offset into the sample, phrase length, pause between phrases, phrase type, output*, and with MAS parameters of *acquiescence* (desire to stay at the location of a food source), and *sociability*. Agents have a histogram of encountered food sources. This histogram affects the decision of an agent's movement. The audio input is recorded into an audio buffer when it is not silent. The existence of sound in the audio input also creates excitement in the virtual environment and the agents start moving at faster rates. There is also communication between the agents. When an agent finds a food source, the agent shares the location of the food source. The agents die if they cannot locate a food source for a certain time. The death agents are reincarnated after a variable duration that is between 5 to 60 seconds. The agents create social networks by sending 'friend requests' and using the sociability ratings. The agents can also leave a network to join a bigger network. Eigenfeldt and Pasquier stated that even when the agents find a food source, and the network becomes static, the social networking still create dynamic behaviors.

Beyls et al. [21] presented a MAS implementation focusing on a cultural phenomena. The system, *earGram Actors* (45) is based on the *Actor* model [19] which is a derivation of the *Party Planner* model [97]. In the actor model, the reactive agents aim to be as close as possible to the agents that they like (and vice versa) to minimize the social stress of the society. The actor model works on two dimension *affinity* and *sensitivity*. Affinity is pre-set and dictates the attraction of an agent towards another. Sensitivity sets a distance threshold to apply affinities of agents. This simple abstraction of a virtual society creates complex movements of agents in the virtual environment. This MAS implementation uses a hybrid audio corpus. The corpus consists of 200 ms long audio samples, and the samples are mapped to a 2D space using dimensionality reduction on a set of audio features, including noisiness, pitch, brightness, spectral width, and sensory dissonance. Then, this 2D audio feature space is mapped to the 2D virtual environment. Hence, the movements of agents in the environment create musical output with the concatenative synthesis.

Likewise, *pMIMACS* (46) is an ecosystemic MAS that generates symbolic music with interpretation [108]. Each agent has the same architecture and has a tune in the memory. The agent with similar tunes performs to each other during each cycle. When an agent performs to another, the

listening agent learns the interpretation of the other agents tune. There are four dimensions of the interpretation in *pMIMACS*: accuracy/tempo, excitation state, key, and microerrors.

Al-Rifaie and Al-Rifaie [1] present a MAS (47) generating musical melodies with symbolic representation. This implementation also uses a swarm intelligence algorithm, *Stochastic Diffusion Search (SDS)* that is inspired by one species of ants, *Leptothorax acervorum*. In SDS, the agents situate in a search space and communicate with one another directly. SDS has two phases: *test* and *diffusion*. During the test phase, each agent implements exploitation. If an agent finds a better solution, the agent is considered *happy*. During the diffusion phase, each agent talks to another agent that is randomly chosen. If an unhappy agent talks to a happy agent, the unhappy agent is considered *lucky*, and the happy agent shares its location with the lucky agent. In each iteration, the number of local unhappy, and lucky agents are stored for the sonification. The focus in this implementation is generating musical scores that sonifies the agent communication. The system uses plain texts as an input, and the letters are mapped to pitch, note duration, and dynamic. The population size is 20, and the number of iterations (episodes) is 10. The authors exemplified this implementation with a melody generated by the text ‘hello music sds welcome to the reality.’

Similarly, Gimenes and Miranda [94] applied cultural ecosystem approach in the design of *Interactive Musical environments (iME)* (48). *iME* concentrates on monophonic melody generation. The system design applies the ideas of “memetics”. Memetics (as in genetics) is the idea that the development of cultural organisms is through the smallest functional units that are *memes* (as in genes). The authors propose the term *ontomemetic* (inspired by ontogenetic) that is ‘the sequence of events involved in the development of individuals musicality.’ The authors also point out the characteristics of ontomemetic systems:

1. Modelling cognitive and perceptive abilities of humans,
2. Using the interaction between artificial entities to create emergency
3. Modelling interactivity through the communication between artificial entities
4. The availability of comparison of different musical styles generated by an application of ontomemetics

iME applies ontomemetics to monophonic melody generation using a feature extraction on MIDI data. The features are melodic direction, melodic leap, inter-onset interval, duration, intensity, and vertical number of notes. *iME* has a virtual ecosystem in which agents listens each other. Each agent has the same architecture. One agent takes the role of *player* whereas the other takes the role of *listener*. The agents have two types of memory: long-term and short-term. The long term memory stores all unique memes that the agent encounters. Each meme has a connection pointer that is the index of the successor meme. Each meme has a weight that increases as the agent encounters a meme more. In that sense, this structure resembles a first-order Markov model. Short-term memory only saves a user-defined number of memes that the agent encountered the latest. The system is capable

to generate music as using solo agents, as well as collective improvisation. The generative algorithm includes a pre-set *compositional and performance map* that guides agents to choose memes.

Our survey of cognitive and reactive musical agents finishes here. We continue by reviewing musical agents that combine cognitive and reactive modules together in their system design.

2.6 Hybrid Musical Agents

Hybrid agent architectures include both reactive and cognitive modules together. Following, we discuss four subcategories of hybrid musical agents.

2.6.1 Hybrid musical agents using statistical sequence modelling

A recurrent theme in hybrid musical agent studies is the implementation of statistical sequence modelling algorithms such as Incremental Parsing (IP), Probabilistic Suffix Trees (PSTs), Factor Oracles (FOs), Partially Observable Markov Decision Processes (POMDP), Variable Markov Models (VMM), Hidden Markov Models (HMM). Many systems presented in this section use Markov Decision Processes or Markov Models or Markov Chains.

Markov Models are finite state machines that encodes patterns of transitions between discrete states using the Markovian assumption. The Markovian assumption of an N^{th} order Markov model is,

$$P(s_t|s_{t-1}, s_{t-2}, \dots, s_1) = P(s_t|s_{t-1}, \dots, s_{\max(t-N,1)}) \quad (2.1)$$

The order of a Markov model dictates how many previous states to be considered to predict and generate the next state. Moreover, the conditional probabilities of the transitions depends on the observed number of transitions between the states.

The environment is discrete and stochastic in Markov Models [160]. The environment is stochastic because given a particular state, the resulted next state of an agent is not certain. Specifically in Markov Decision Processes, we can define a probability function $p(s'|s, a)$ where s is the current state of an agent, and a is an action. *Reward* function, $r(s, a)$, evaluates an action, a , performed in a particular state, s . *Policy* (or *decision rule*), d , is an assignment function, $d : S \rightarrow A$, where S is the set of all possible states and A is the set of all actions that are available to an agent. Hence, a *policy* specifies which actions should be performed in which states. In Markov Models, *value iteration* algorithm defines how to find an optimal policy [160]. Moreover, Markov Models can apply learning. Agents generate the transition probabilities between states during learning. For example, Martin et al. [121] implement Partially Observable Markovian Decision Processes (POMDP) (49) in the design of a hybrid musical agent that listens to human performers (the system (74)). The system generates melodies in the key of the human performer using tonal harmony theory in music.

The initial experiments on using statistical sequence modelling algorithms for music focused on how to create optimal tree form representations using compression algorithms for online applications of music. Two early works [61, 9] concentrated on generating sequences of melodies using

Lempel-Ziv compression algorithm. Later, their work evolved into two interactive musical agents, the Continuator and *OMax*. The Continuator (50) is a well-known musical agent on the problem of musical style *imitation* [153, 152]. The *Continuator* study proposes the MuMe problem of *continuation*, that is, continuing a performance in the style of the performer when the performer stops. Using symbolic representation of music (MIDI), Pachet introduces hierarchy and bias to Variable-Order Markov Models to handle the polyphony, noise, and arbitrary rhythmic structures in the input. The agent architecture consists of two parts: *analysis* and *generator*. The analysis module has three submodules: *phrase end detector* (adaptive temporal threshold mechanism), *pattern analyzer* (the generation of Variable-Order Markov Model), and *global property analyzer* (number of notes per second, tempo, meter, and overall dynamics). The Continuator has two modes of interaction: *question and answer*, and *collaboration*. What Pachet refers as question and answer is *call and response*, a well-known improvisation setting in the context of jazz. In the collaboration mode, the *Continuator* implements accompaniment by listening to a human performer in real-time and adapting the generated output in parallel with the human performer's style. Moreover, Pachet [153] proposes three implementations of *Continuator*. First, a musician can play with a Continuator trained on a famous musician's performance. Second, multiple musicians can have multiple Continutors trained on different musical performances. Musicians can also have Continutors trained on the same corpus. Third, the *Continuator* can extend a soloist's capability or accompany a soloist by training on a corpus of chord sequences.

Beatback (51) [101] also focuses on the tasks of accompaniment, and call and response by implementing Variable Order Markov Models to generate musical rhythms. The system represents rhythm sequences in three dimensions: inter-onset time difference, velocity (MIDI), and drum type (instrument). *BeatBack* focuses on two musical applications: accompaniment, and call and response. Hawryshkewich et al. present a technique called *Drum-kit Zoning* to use *BeatBack* as an *Expanded Instrument System* that expands the performance of a drummer. The pattern generation in *BeatBack* has two modes: *query* and *build*. In the Query mode, *Beatback* uses the last rhythmic pattern of its input to search for and assign probabilities to possible next patterns. In the Build mode, *Beatback* generates a rhythmic pattern using the probabilities generated by the Query mode.

Ringomatic (52) is another musical agent that generates rhythm accompaniment [12]. The authors implement two classification tasks to automatically generate a hybrid corpus. The first task is to find solo drum sections in recordings and the second task is to label them with three energy levels of low, medium, and high (Figure 2.14). *Ringomatic*'s architecture includes constraint-based concatenative synthesis to generate audio. Ringomatic sets the constraints of energy, onset density, and pitch. The authors propose a new technique called incremental adaptive search that is an implementation of local search techniques in constraint satisfaction problem. The constraints are introduced to the system as cost functions. There are two types of constraints: local and global. Local constraints look only at the current state to predict a next state whereas global constraints include past states in the cost function. Aucouturier and Pachet 2005 also present an analysis of the system in a duo with a human performer playing MIDI keyboard.

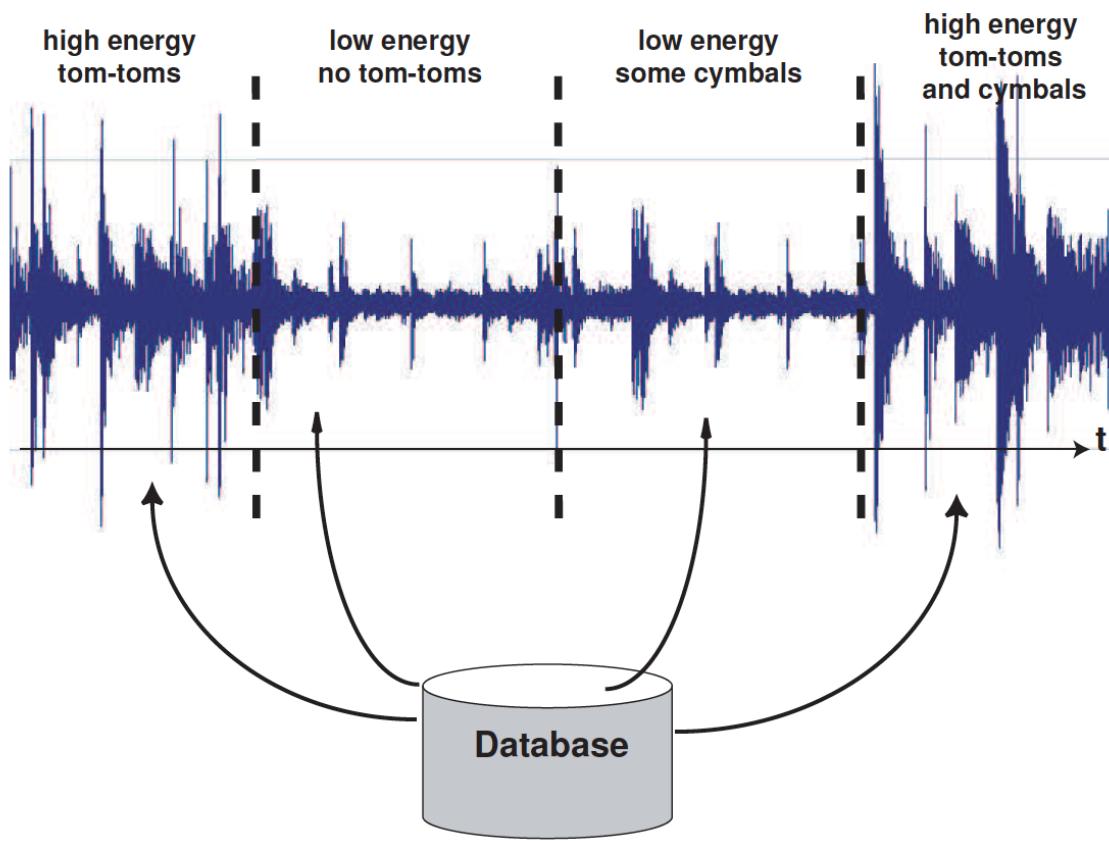


Figure 2.14: Energy-based generation in Ringomatic [12]

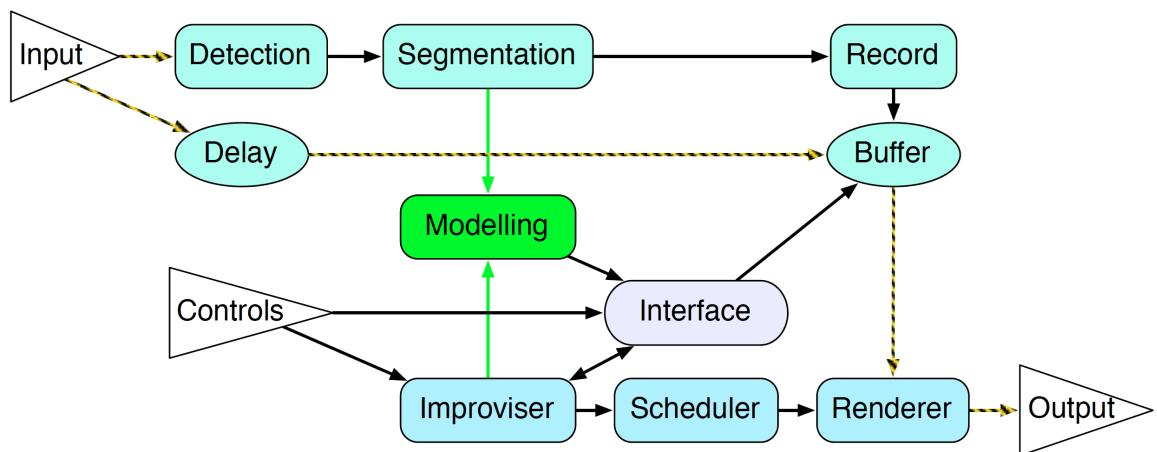


Figure 2.15: The block diagram of OMAX [115]

Factor Oracle (FO) is a finite state automata that is a variation the suffix tree. FO represents substrings and patterns in a sequence, that is, at least all *factors* of a sequence. FO has three types of links: internal links, external links, and suffix links. Internal links are forward links between successive states. External links are forward links that jump longer than successive states. Suffix links are backward links that point the longest repeating factor in the previous states. FO allows incremental learning, and learning is linear in time and space [113]. Assayag and Dubnov [8] compare IP, PSTs, and FOs for the symbolic sequences of music. Assayag and Dubnov conclude that FOs suit the best to satisfy incremental and fast online learning, time-bounded generation of musical sequences, and implementation of multi-attribute models to deal with the multi-dimensionality of music. Within the last two decades, many studies implemented FOs in musical agents [8, 10, 63, 85, 115, 146, 174, 150, 86, 192, 59, 118, 80, 149, 188].

Assayag et al. [10] present a framework to implement musical agents with FOs, called *OMAX* (54). OMAX uses a FO based real-time machine improviser scheme [8]. Assayag et al. also propose two unique implementations of FOs: one with reinforcement learning, and the other with meta-level learning. First mentioned by Dubnov and Assayag [60], *OMAX* listens to a performer, and learns the style of the performer using FO. The problem of *style imitation* is well-known in MuMe field [155]. The agent generates by using navigation strategies combined with the links and factors within the FO model. browsing the model using diverse navigation stratégies, and renders these sequences sonically. Thus, the generated material is recombination of musical material in the agent's memory. The agent utilizes polyphonic pitch duration slices with MIDI for the musical applications with symbolic representations, and real-time recorded audio segments in the case of audio [115]. Lévy et al. implement *OMAX* in MAX 5, including pitch estimation, and spectral clustering with Mel Frequency Cepstral Coefficients (MFCCs) and Fast Fourier Transform (FFT) in the analysis module. Ongoing artistic use of *OMAX* appears in two context, duo with a human performer playing an acoustic instrument, and control of an *OMAX* musical agent by an electronic musician [115]. The I/O of *OMAX* can be symbolic representation of music, or audio signals, or video signals [27, 115].

The hybrid musical agent architecture (55) of Cont et al. [55] also implements FOs with an anticipatory model of musical style imitation with collaborative and competitive reinforcement learning. The authors use *multiple viewpoints* [54], and there are four factor oracles trained for musical dimensions of *pitch*, *pitch contour*, *duration*, and *duration ratio*. The system can be used in two modes: *interaction* and *self-listening*. In the interaction mode, the agent listens to another agent (or human performer) whereas the agent listens to its own audio output in the self-listening mode.

Collins [49] also presents a musical agent, called *Improvagent* (56) that uses reinforcement learning with symbolic representations of music. Using the MIDI input, the agent computes a set of onset, pitch, and rhythm features as well as higher level features such as key, pitch class, expressiveness, and density. *Improvagent* treats input frames as the states of the environment. The system clusters environment states using k-nearest neighbors with Euclidian distance. The agent also updates its

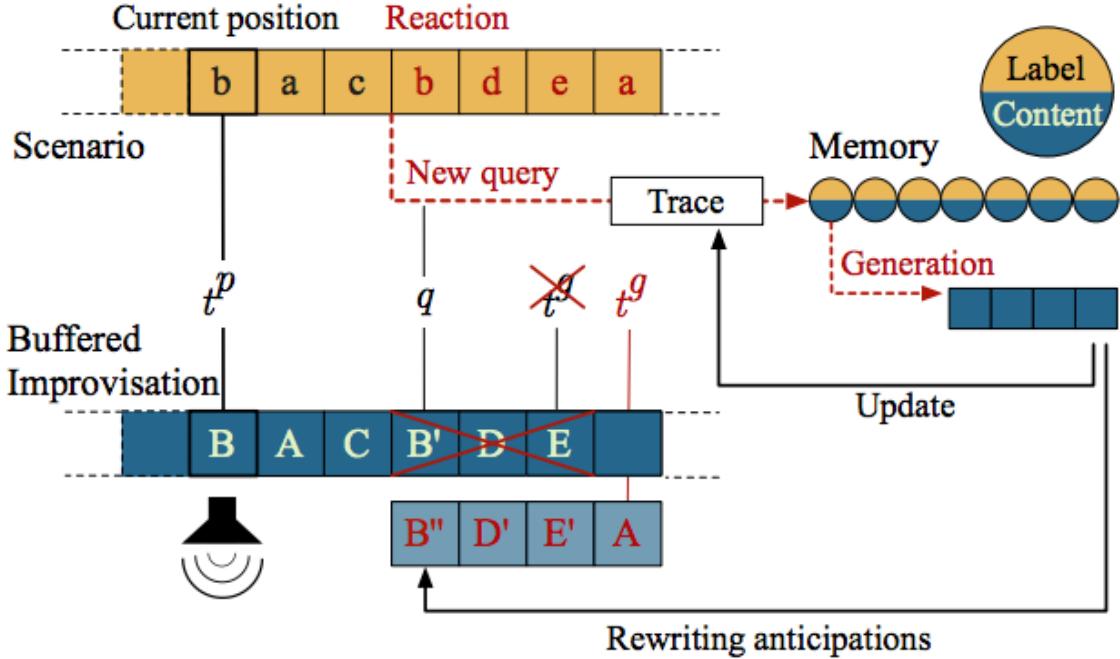


Figure 2.16: The improvisation renderer in Improtak [148]

database in real-time. The included reinforcement algorithm is Sarva¹¹. Improvagent generates the audio using the concatenative synthesis.

Improtak [146, 147, 148, 149] builds upon OMAX, and implements *OMAX* with an introduction of tempo, beats, harmonization, and arrangement. *Improtak* system uses three Factor Oracles for improvisation, harmonization, and arrangement, using the symbolic representation of music (MIDI) for the applications of improvisation and accompaniment. Nika et al. [149] further develop *Improtak* by introducing a *scenario/memory generation* model. Figure 2.16 shows the guided improvisation in *Improtak* with a scenario and memory. The authors use any alphabet in the musical context, such as audio, MIDI, or sound synthesis parameters. A symbolic sequence of labels defined with the alphabet is the *scenario* whereas a sequence of musical contents labelled using the alphabet is the *memory*. The improvisation is guided by the scenario using two strategies: *anticipation*, and *digression*. Using anticipation, *Improtak* searches the memory for a starting sequence. The constraint is that the starting sequence matches the future labels that follows the current state of the scenario. Digression strategy ensures that *Improtak* finds a continuation sequence in the memory. This continuation sequence matches both past and future states of the current state of the scenario. The implementation consists of three agents: *improvisation handler*, *dynamic score*, and *improvisation renderer*. The improvisation handler is a reactive agent that implements the guided music generation using a scenario and a memory. The dynamic score handles perception of *Improtak*'s environment.

¹¹The details of Sarva is available in the book on reinforcement learning by Sutton and Barto [175]

The improvisation renderer conducts the output generation using the content generated by the improvisation handler. Nika et al. [149] points out two cases of performance with Improtok: human and machine, and machine only. When Improtok is trained on audio, the agent conducts online audio generation using a phase-vocoder. Hence, Improtok can sample live audio and apply time stretching, pitch-shifting, and crossfade transformations in real-time to temporally and harmonically align the generated improvisation with a pre-defined scenario.

In parallel with the studies on *OMAX* framework, Dubnov et al. [63, 64] present *Audio Oracle* (58). Inspired by FOs, *Audio Oracle* is an algorithm that detects repeating sub-clips of variable length in audio data. Dubnov et al. define these sub-clips as *audio factors*. Similar to FO, *Audio Oracle* analyses an audio file as a string of audio feature vectors. The user can choose different audio features (or combinations of audio features) to train an *Audio Oracle*. Forward links in *Audio Oracle* refers to the states that generate similar patterns by continuing forward whereas backward links correspond to the states sharing the largest similar sub-clip in an audio file. *Audio Oracle* uses Euclidian distance between audio features to decide if two states belong to the same class. The user sets a similarity threshold, and if the Euclidian distance between two states is below the threshold, those states are accepted as equivalent. High similarity threshold means that distant states are more likely to be labelled with the same class, thus decreasing the size of the alphabet. Furthermore, Dubnov et al. [64] introduce automatic threshold selection for the *Audio Oracle* using the notion of *Information Rate (IR)* in Signal Processing [62]. AO uses the threshold that gives the highest information rate.

Surges and Dubnov [174] further developed *Audio Oracle* studies by introducing a system for music analysis and machine improvisation, called *PyOracle* (59). Similar to *Audio Oracle*, *PyOracle* includes an off-line learning that inherits signal complexity and familiarity analysis. Surges and Dubnov relate complexity and familiarity to aesthetic appreciation with Birkhoff's idea of aesthetic measure [161]. Birkoff defines aesthetic measure as the ratio between the order and the complexity. *Audio Oracle* uses IR to balance between the order and the complexity. IR measures the reduction of a signal's uncertainty using signal's past values. Surges and Dubnov stated that low IR refers to higher complexity and lower order whereas high IR corresponds to lower complexity and higher order. *Audio Oracle* uses IR measure to set the uniqueness distance threshold between the states. Surges and Dubnov aim for the highest IR in the implementations to extract the musical form information of a signal during the PO's learning process.

Building on the previous studies of FOs, *Audio Oracle*, and PO; Wang and Dubnov [192]; and Arias et al. [7] introduce another musical agent system using *Variable Markov Oracles* (VMO) (60). VMO allows adaptive symbolization of audio features to provide representation of higher musical structures. The system implements *Petri Net* graphical language for concurrent and distributed system design, *PyOracle* to create *Audio Oracles*, and I-score [13] to control generated models with graphic scores. The authors mention that the previous studies on musical implementation of Factor Oracles have been criticized by not representing higher musical structures, and this study addresses the representation of higher musical structures using *Petri Net*. Arias et al. propose that a possi-

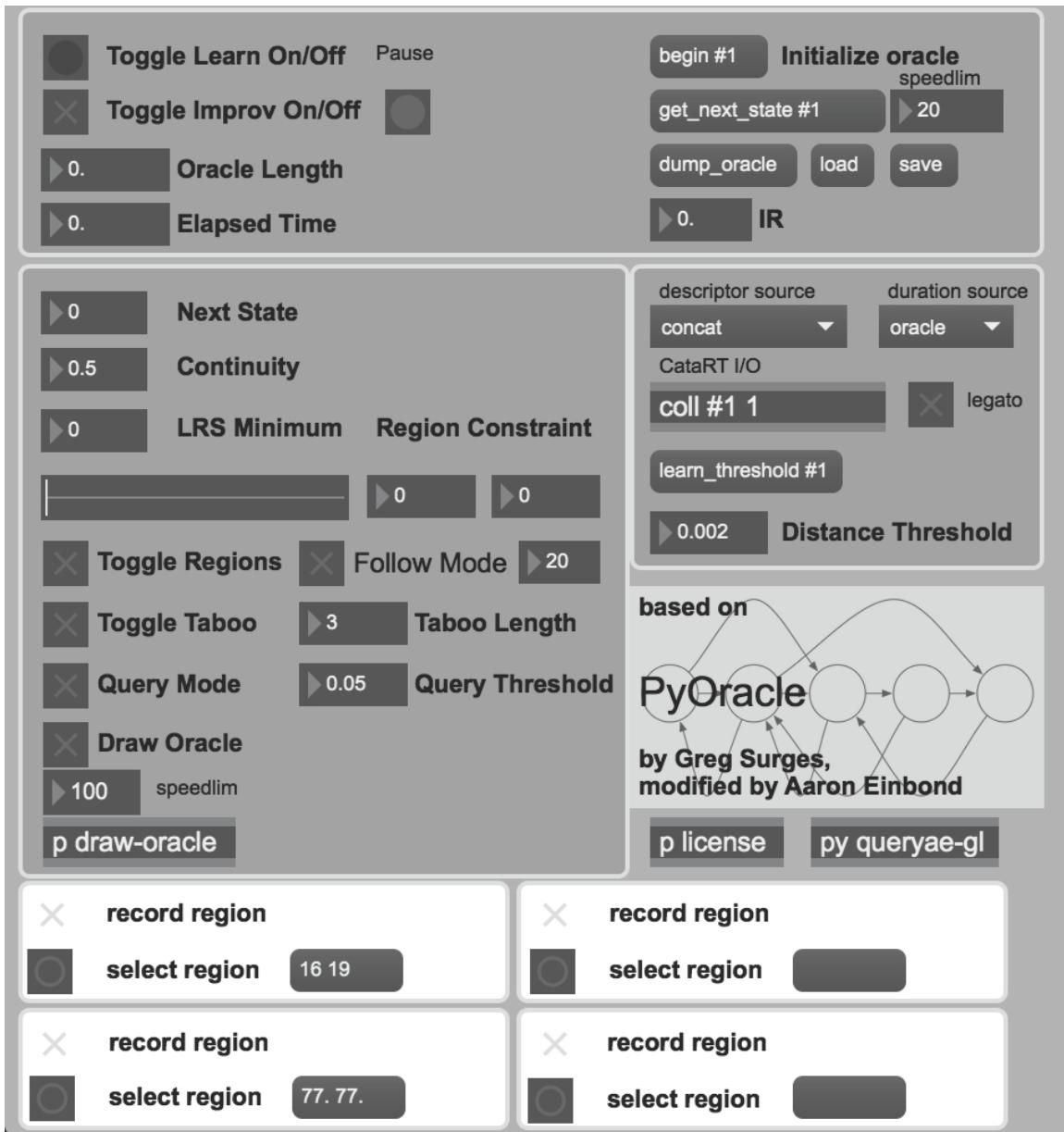


Figure 2.17: The interface of PyOracle

ble next step for the development of this system is the introduction of *scenario/memory generation* model, presented by Nika et al. [148].

Freely Improvising, Learning and Transforming Evolutionary Recombination (FILTER) [150] is a musical agent that combines FO and a type of Markov Models, called Hidden Markov Models (HMMs). HMMs are widely used to model temporal discrete sequences. HMMs consist of hidden states and observed states. The number of hidden states can be different than the number of observed states. The transition matrix is the likelihood of transitions between hidden states. The observation (or emission, or confusion) matrix is the likelihood of observations given a hidden state. HMMs have three applications: evaluation (likelihood of an observed sequence given an HMM), decoding (the sequence of hidden states that most likely to generate the observed sequence), and learning (generating a HMM given a sequence of observed states).

The musical application of *FILTER* is free improvisation. *FILTER* implements style imitation based on unsupervised learning. The learning applies Smalley's approach on textures and gestures in Electro-acoustic Music [170]. Using an inter-onset threshold, *FILTER* samples the audio input of the last N seconds and the memory encodes the temporal changes of audio features. If the recorded sample is dissimilar to the anything in the current memory, it is added to system's memory. *FILTER* includes sonic gesture and texture analysis. *FILTER* applies continuous gesture recognition method proposed by Bevilacqua et al. [15] to learn sonic gestures of the input using Linear Predictive Coding (LPC), Mel-Frequency Cepstral Coefficients (MFCCs), autocorrelation coefficients, and YIN algorithm features (frequency, energy, and periodicity). The gesture recognition algorithm combines HMM with dynamic time warping. The system can learn a dictionary of gestures either offline using a corpus or online by listening to the input. The gesture recognition algorithm outputs the likelihood of gestures. Using the likelihood, *FILTER* can perceives the level of deviation from the current gesture of the input. The system also inherits a non-linear time-frequency analysis called intrinsic mode function to comprehend the sonic texture of the input. *FILTER* includes two types of memory: *semantic* and *episodic*. The semantic memory is the dictionary of distinct gestures whereas the episodic memory applies FO to learn temporal structures of the input. *FILTER* also applies a mutation only Genetic Algorithm (GA) for the adaptive goal decision process. The system introduces adaptivity by mapping the gesture/texture likelihood values to the fitness of GA.

Lastly, *SpeakeSystem* (62) is a musical agent with Variable Markov models (VMMs) [203]. The agent uses FM synthesizer to generate audio. The modulation index of the synthesizer changes as the length of sequences generated by VMMs varies. The authors stated that using two VMMs, where one VMM handles the rhythm and the other focuses on the pitch, generates more varied output, comparing to the case with one VMM.

2.6.2 Hybrid musical agents combining statistical sequence modelling with rule-based models

Martin et al. [122] present a framework for non-technical users to design musical agents. This framework, called *The Agent Design Toolkit* (ADTK) (63) implements the ideas of interactive ma-

chine learning in musical agents. ADTK consists of three elements: a set of recorded performance variables, a set of probabilistic temporal models, and a set of rules defining the relation between performance variables. The framework uses VMMs for probabilistic temporal models and association rule learning (ARL) algorithms for automatic rule generation using the recorded performances. Following this study, Martin et al. [124] introduce ADTK to Ableton Live, a well known Digital Audio Workstation (DAW). Martin et al. conducted two case studies on ADTK, designing a musical agent that improvises electro-acoustic music, and a musical agent generating *Drum and Bass* music. Martin et al. [123] mention a possible computational complexity problem with the initial versions of ADTK. The automatic rule generation solves the constrained satisfaction problem using ARL algorithms. However, it is not possible to know how long ARL algorithms take, and how many solutions the algorithm produces. This makes the systems designed with ADTK framework susceptible to bugs in real-time performances. To address this computational complexity problem, Martin et al. proposes binary decision diagrams (BDDs). Although the introduction of BDDs does not completely solve the computational complexity problem, the designer can examine if an agent is capable of real-time performance before the performance. Thereby, the system is no longer susceptible to bugs in real-time performances. Martin et al. also compare BDDs-based ADTK to the initial version of ADTK with ARL, concluding that the parameter update duration was more predictable in BDDs-based implementation than the ARL-based implementation. Martin and Bown [120] also demonstrate ADTK on style imitation. Bown and Martin [33] mentioned that the musicians could control the agents designed using ADTK. Hence, these agents stand somewhere between an extended instrument and an autonomous performer.

CinBalada [64] is another multi-agent system that combines statistical sequence modelling with rule-based models [167]. The system generates polyphonic rhythmic sequences as the symbolic representation of music. Sampaio et al. are inspired by the music styles with an emphasis on the rhythm such as taiko, pungmul, samba batucadas, and maracatu. *CinBalada* includes three rhythm representations of Time Unit Box System, Polygonal Representation, Time Elements Displayed as Squares to calculate rhythmic measures of offbeat-ness, evenness, and rhythmic similarity as chromatic distance. *CinBalada* includes these rhythmic measures in the evaluation functions. *Cinbalada* is a homogeneous MAS with multiple roles. There are multiple *rhythmic roles* that each agent can choose. The number and the type of rhythmic roles depend on the implemented musical style. For example, a Batacuda implementation has three roles of *base*, *complementary base*, and *solo*. The evaluation functions also change depending on the implemented style. Within a bar, agents in *Cin-Balada* negotiate what to play in the following bar. The agents share their rhythmic patterns with the other agents. *CinBalada* outputs only the patterns that score the highest on the evaluation functions.

2.6.3 Hybrid musical agents with Artificial Neural Networks

Artificial Neural Networks (ANN) is a set of Machine Learning algorithms. ANN algorithms are inspired by the theories of neuron activation and sensory data processing of neural systems in na-

● Line affected

○ Line read

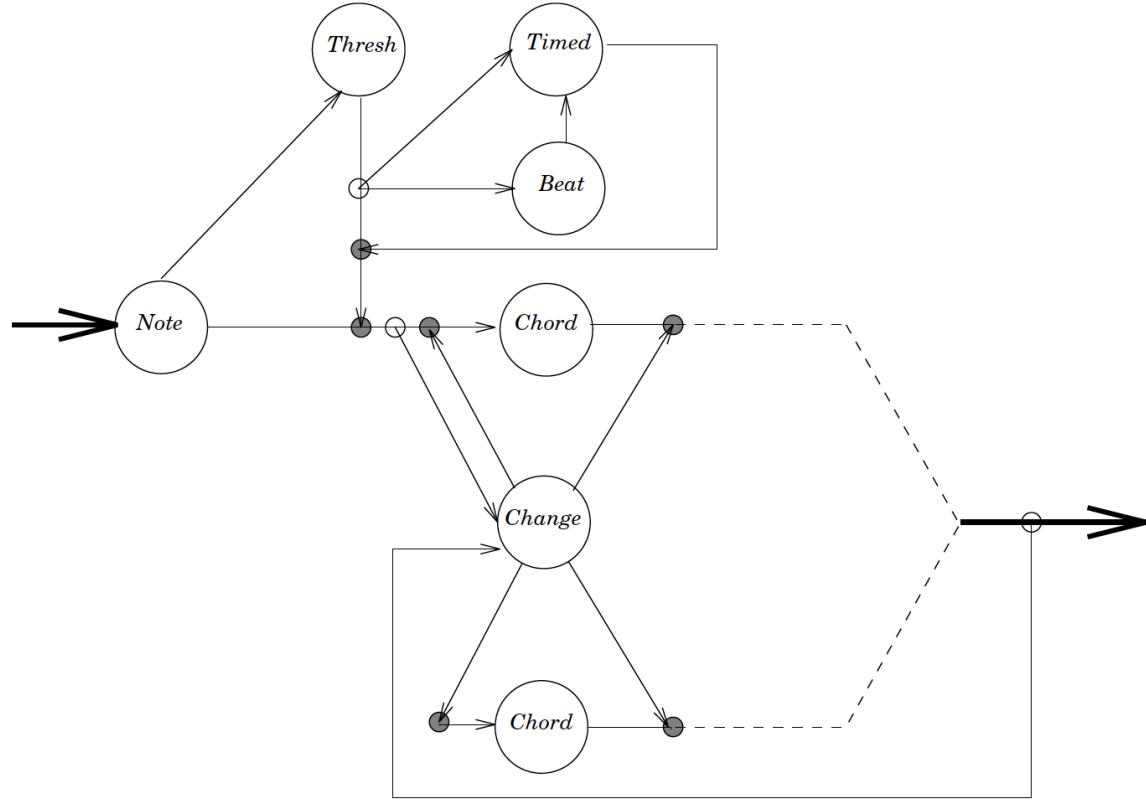


Figure 2.18: The Subsumption architecture of the Reactive Accompanist [42]

ture. ANN has been applied to the Machine Learning problems of classification and linear regression [136].

The *Reactive Accompanist* (65) is the first musical agent system that implements Subsumption architecture, including three ANNs in different layers [42]. Subsumption architecture (mentioned in Section 2.5) implements a hierarchical set of rules in which lower layers have higher priority, or vice versa. Reactive Accompanist is a mono-agent system with audio input and symbolic output (MIDI). There are three layers in the architecture: *pitch*, *chord*, and *time*; ordered from the lowest to the highest layer respectively (Figure 2.18). The pitch layer has two modules. The first one implements Fourier transform and outputs frequency-gain pairs. The second module is an ANN with supervised learning. The input is frequency-gain pairs whereas the output is pitch classes. The chord layer is also an ANN. The input is pitch classes and the output is predefined chords. The highest layer in the hierarchy, time has four modules of *thresh*, *beat*, *timed*, and *change*. The first three modules handle rhythm. Beat module implements tempo estimation with ANN and change module handles chord changes. The application of this system is accompaniment of live input. Bryson implemented the system in the first half of 90s when Fourier transform calculation was still too computationally com-

plex for online applications. Because of the Fourier transform calculation in the pitch estimation, this system works offline.

Another musical agent with ANNs is *NN music* (66) [205]. The architecture is an implementation of the musical agent framework, *PQf* proposed by Blackwell et al. [26], where P implements listening and analysis, Q handles performing/synthesis, and f conducts patterning, reasoning, or generative functions. *NN music* includes two analysis functions in P module: *parameterization of pitch characteristics* and *statistical representation of musical behaviour*. These two analysis output two independent state representations: a set of recently identified pitches and a set of statistical representation of audio features computed over 50 ms audio frames. The statistics are calculated with a varying window of 5 to 30 seconds. *NN music* includes two Multi-layer Perceptron (MLP) neural networks that are connected in series. Both networks are trained with the back propagation algorithm and have three hidden layers. The statistics of audio features is the input of the first ANN. The second MLP maps the classification output of the first MLP to synthesis parameters. The second MLP outputs a set of synthesis parameters with a probability distribution. Hence, the synthesis module inherits stochastic behaviors. The training of the first MLP is ongoing during the performance. There is a similarity algorithm in the system that checks if the current state is similar to the states that are used in the training. If the current state is not similar, ANNs are trained with the current state. The second MLP is trained before the performance.

Bown [32] presents a musical agent (67) with continuous-time recurrent neural networks (CTRNNs). Each node is connected to each other with a directional weighted connection (synapses) in CTRNNs. In this implementation, the nodes have sigmoid activation function to process the directional weighted outputs of previous neurons. CTRNN is a blackbox type module with N (and M) floating point input (and output) values. In addition to the learning in CTRNN, Bown implements a mutation-only Genetic Algorithm (GA) that evolves multiple CTRNNs in parallel. Bown mentions that the GA includes a multi-objective fitness function that evaluates CTRNNs' 'success at acting with the responsive properties of dynamic reservoirs' and success at showing repetitive behaviors when the input is repetitive. This musical agent maps the CTRNN output to continuous synthesis parameters as well as the decision of triggering sound events. This agent has been presented in many concerts, performing with human-performers playing trombone, clarinet, and shakuhachi. Building on this agent, Bown presents another musical agent that includes Decision Trees (DTs). Bown states five advantages of DTs over CTRNNs: discrete output on each time step, the ease in the analysis of the agent's behavior, efficiency, and adaptive self-calibration of decision boundaries. This implementation with DTs also includes a mutation-only GA evolving multiple DTs in parallel. The fitness function is single-objective, and it is for maximizing the number of DT leaf nodes visited. This agent has also been presented in many venues, with human-performers playing trumpet, bass clarinet, and electronics.

Kohonen Network is an application of ANN [111]. Although Kohonen networks, including Self-Organizing Maps (SOMs), are proposed in the early 1980s [110], it is recently discovered by studies related to the MuMe field. Smith and Garnett [172] present a musical agent (68) with adaptive

resonance theory (ART) and reinforcement learning (the system number 69 in Table 2.1). This implementation focuses on monophonic melody generation. The ART network is a self-organizing neural network for classification and categorization of data vectors. ART network differs from SOM in training. Each input vector updates only one node in the ART network whereas in SOM, a set of nodes are updated. The agent converts the MIDI input to a combined feature vector of *pitch class*, *interval*, *interval and direction window*, *direction sign*, *octave*, and *interval octaves*. The reinforcement learning implements two functions. First, when the agent updates an existing node in ART network, the agent calculates the reward using the previous and the updated state of the node. Second, when the agent creates a new node in ART network, the agent calculates the reward using a user set parameter called *vigilance*. The authors also present two examples of the agent on free improvisation. Another example presents the output of an agent trained using J.S. Bach's six unaccompanied cello suites.

Smith and Deal [171] present a musical agent application (69) of SOMs. This agent architecture utilizes chroma audio feature extraction in the perception stage to extract the pitch and rhythm information from the agent's audio input. Then, the agent's memory organizes extracted chroma vectors in two levels, long-term and short-term. The authors introduce adaptive behavior to the agent's short-term memory by using a SOM. In this system, training of the SOM is continuous. The decision module of this agent calculates a measure of learning in the SOM using the difference between the previous state and the trained state. Smith and Deal state that this learning measure is analogous to the Kolmogorov complexity. The decision module targets a learning rate. This agent follows its input if it is complex enough to satisfy the target learning rate. If not, the agent diverges from the input to increase the overall complexity. Hence, the decision module provides SOM a distance to the audio input vector. Then, the agent decides on a corresponding SOM node. This node is the input vector of the long-term memory. The long-term memory uses a k-d tree to search in a multi-dimensional space. Each vector provided from SOM is a search query to locate the closest vector in the long-term memory. The long-term memory consists of pre-defined audio files, and the agent does not update the long-term memory.

Martins and Miranda [125] present a musical agent (70) with *SARDNET* generating rhythms. *SARDNET* is a variation of SOMs with an addition of temporality. *SARDNET* deals with event sequences using node activation values and differs from SOMs in two ways. First, the winning neurons are not included in the subsequent training. Second, the activation values of each node are decreased in each step. *SARDNET* represents the input sequence as all active nodes ordered by their activation values. Martins and Miranda approach the rhythmic events as three-dimensional events. These three dimensions are timbre, velocity, and inter-onset interval. The musical agent includes two ANN cascaded in series. Symbolic representation rhythmic phrases (MIDI sequences) are the temporal input of the *SARDNET*. The output of *SARDNET* is connected to a one-layer Perceptron with three outputs. The training of this musical agent is through pre-recorded rhythms. The authors mention that after fifty iterations, the agent starts to self-organize. Notice that we also encountered the idea of evolving rhythms in Eigenfeldt's [70] Kinectic Engine (see Section 2.5.1).

2.6.4 Hybrid Musical Agents with cognitive models

Camurri et al. [44] present a musical MAS framework called *Hybrid Action Representation and Planning (HARP)* (71). Using the idea of graphical visual programming, *HARP* provides flexible programming environment to the user. The system is capable to create a hybrid agent system. The framework is inspired by MAS implementations in Robotics. The application of this framework is assisted composition, performance, and analysis. The authors define two main components of *HARP*: *symbolic* and *sub-symbolic*. Symbolic components implement compositional syntax and semantics, including domain specific knowledge representations. Sub-symbolic components are the reactive modules of the system with a network of cooperative agents. Sub-symbolic components process the signals of MIDI, audio, or visuals. The authors also give an example of a theatre performance in which *HARP* framework is used to program a software controlling sound, music, and three-dimensional computer animation of humanoid figures interacting with real actors on stage.

The hybrid musical agent architecture (55) of Cont et al. [55], mentioned in Section 2.6.1, explores musical agent applications using the mental representations of expectation in the problem of style imitation. There are four types of mental representations of expectation, proposed in the literature of psychology of musical expectation: veridical expectation (expectation of familiar works, related to episodic memory), schematic expectation (related to the semantic memory), dynamic adaptive expectation (related to the short-term memory), conscious expectation (related to the conscious reflection and prediction) [105]. Cont et al. apply these ideas to MuMe by using an anticipatory model of musical style imitation with collaborative and competitive reinforcement learning. Within four types of anticipation (*Implicit*, *Payoff*, *Sensory*, and *State*), the authors implement payoff, and state anticipation models.

Similarly, Gifford and Brown [92] focus on the idea of using anticipatory timing to plan future actions of a musical agent. The system, called *Jambot* (72) is a hybrid musical agent that generates percussive musical rhythms. *Jambot* can generate rhythms by listening to other performers, or alone. The authors define anticipatory timing as a search for the best next note and when to play this note. The study stated that anticipatory timing enhanced greedy search while slightly increasing the computational complexity. *Jambot* includes a fitness function that evaluates possible actions and possible acting times for the next action. *Jambot* repeats the fitness evaluation on each time frame (audio frame). Gifford and Brown also present examples of system's output with and without anticipatory timing.

In the later versions of *Jambot*, Gifford [91] introduces musical expectation in their hybrid musical agent. *Jambot*'s application is percussive accompaniment to a live audio input. The authors are inspired by the previous works on musical expectation and propose metre as a framework for musical expectation. The system design involves metrical ambiguity to balance novelty and coherence. *Jambot*'s architecture has three modes that controls level of metric ambiguity: disambiguation (use only the most plausible meter), ambiguation (use all plausible meters with equal weights) and following (use all plausible meters with the weights adjusted by plausibility). The reactive behaviours

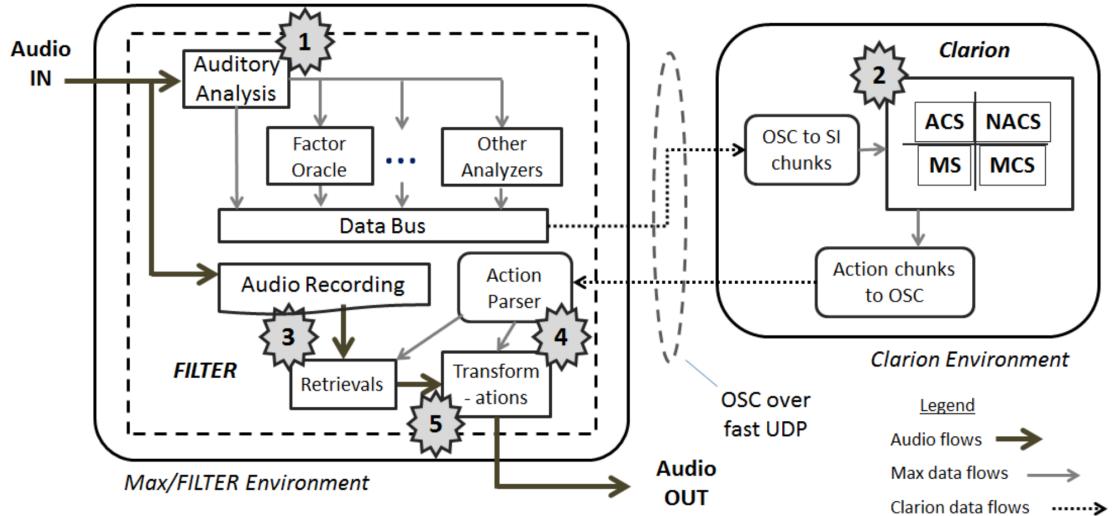


Figure 2.19: The architecture of Mocking-bird [118]

in this system includes three approaches to fluctuate between imitative and intelligent actions: '(i) mode switching based on confidence of understanding, (ii) filtering and elaborations of imitative actions, (iii) measured deviation from imitative action according to a salient parametrisation of the action space.'

Another recurrent theme in cognitive musical agents is motivation-driven, goal oriented musical agent architectures [17, 18, 118]. Beyls [17, 18] focuses on the motivation-driven musical agents (73). The architecture includes two-dimensional space (stability versus introverted-extroverted) to model behavioural changes. The system abstracts motivations as two types of drives: *integration* and *expression*. Integration drives aim to follow the input data whereas expression drives seek to move away from the input data. The *compound function* sets the drive of the agent depending on the levels of integration and expression. This hybrid musical agent implementation also includes reactive modules with an implementation of reinforcement learning and a Genetic Algorithm module (see 2.5.1) that evolves drives. Beyls also analyzed the agent and shows that the fitness function of GA successfully follows the drives set by the compound function.

Lynch's [118] work is the only study that uses a cognitive architecture (CLARION) presented in the Cognitive Science. The system, called *Mocking-bird* (74) combines Van Nort's FILTER system with the *Clarion* cognitive architecture. The *Clarion* cognitive architecture consists of four sub-systems, that are the *Action Control System (ACS)*, *Non-Action Control System (NACS)*, *Metacognition System (MCS)*, and *Motivation System (MS)*. Users can implement either the complete Clarion architecture or any number of its sub-systems. The *Clarion* architecture decides the actions of *Mocking-bird*. These actions indicate a pre-recorded sample to be played starting from a point with a duration, and post processing effects such as pitch shift and time stretch.

The last four systems that we survey includes Affective Computing. First, *MAgentA* (75) is a cognitive musical agent that focuses on generating "film like music" for games using an algorithm

database with affective labels [45]. *MAgentA* is a part of the game framework *FantasyA* in which the user can influence the affective state of the characters they play [154]. *MAgentA*'s architecture has three modules: *perception*, *reasoning*, and *action*. This architecture resembles Blackwell et al.'s [26] PQf musical agent framework. Perception module checks the affective state of the environment and generates outputs when the affective state changes. Reasoning module checks if the new affective state can be generated with one of the algorithms in the database. If not, the exception handling module uses the history database to decide the most appropriate algorithm to use. Once the agent decides which algorithm to use, it sends the algorithm to the composition engine. The action module generates the audio output using data coming from the composition engine.

Second, Dubnov and Assayag [60] combine the flow model with Factor Oracle and create a musical agent (76) within the *OMAX* framework. The agent listens to other performers online to train the FO. The flow model defines the notion of Experience Flow that explores the relationship of mental states with the an activity where a subject is fully engaged and immersed with the tasks [56]. Dubnov and Assayag changes the original flow model dimensions, *challenge* and *skill*, with two dimensions of *emotional* and *familiar*, and 8 categories of *arousal*, *flow*, *control*, *boredom*, *relax*, *apathy*, *worry*, and *anxiety*. The authors mapped these two dimensions of flow model variation to the *replication*, *innovation*, and *recombination* parameters of their musical agents. These parameters controls the probabilities of links within the FO.

Third, Kirke and Miranda [109] implemented Affective Computing with a virtual ecosystem. We mention other ecosystemic approaches in musical agents in Section 2.5.2. The application of the system is melody generation for assisted composition. The system is called *Multi-agent Affective Social Composition System (MASC)* (77) and combines Affective Computing with a MAS. The application in focus is assisted composition. MASC generates melodies through communication and artificial emotional influence between agents. This system implements affect estimation of musical melodies with continuous two-dimensional affective space. The dimensions are *valence* and *arousal*. This 2D model is common in Affective Computing in sound and music [66]. The agents situate in a virtual environment. The number of agents is in the range of 2 to 16. Each agent has a monophonic melody. The agents share their melodies with each other. Agents learn other agents melodies if the emotional state of the melody is close to the agent's emotional state. Moreover, the emotional states of agents are also affected by emotional states of other agents during communication. The authors present examples of melodies generated by this system. Also, the first author shared his compositions in which the first author used this system to generate melodies to assist the composition process.

The last system that apply Affective Computing is *Musical Agent based on Self-Organizing Maps (MASOM)* [177, 179]. *MASOM* is a machine improvisation architecture for live performance (Figure 2.20). The musical context of *MASOM* is experimental music and free improvisation. *MASOM* is a flexible agent that can be trained on any audio file such as a recording of a performance or composition. *MASOM* extracts the musical form of an audio file using unsupervised learning. The learning stage has four steps. First, *MASOM* segments the audio file using the multi-granular

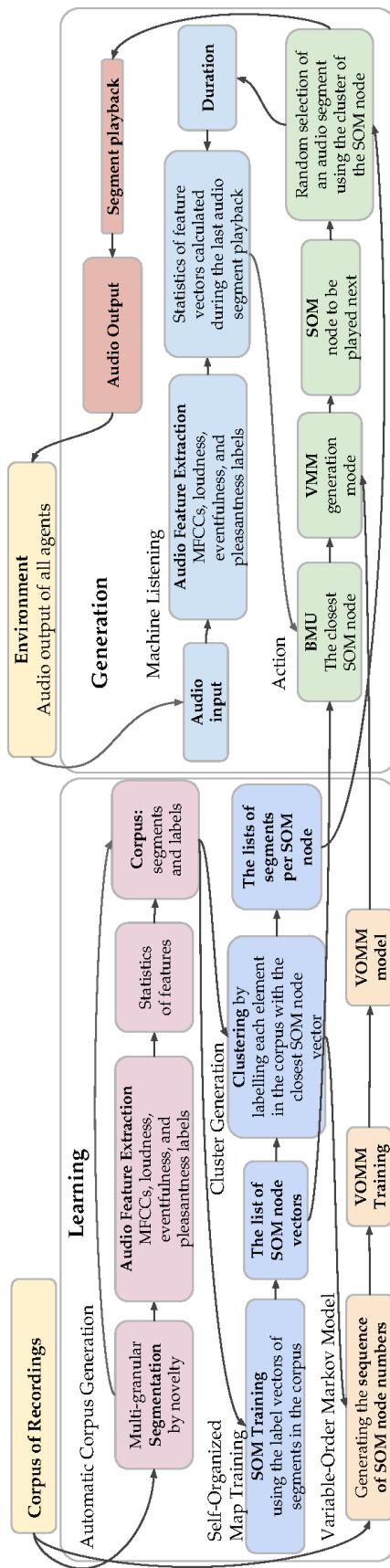


Figure 2.20: The system architecture of *MASOM* [177]

segmentation [112]. Multi-granular segmentation uses novelty curve to segment an audio file. Second, each audio segment is labelled with duration, eventfulness, pleasantness, and timbre features. Third, *MASOM* uses SOM to cluster these audio segments. The last step of the learning stage is VMM training. Each segment of the original audio file is labelled with the closest SOM vector in the feature space. Using the order of segments in the original audio file, *MASOM* generates a string of SOM nodes. This string represents the musical form of the original audio. VMM is trained using this string of SOM nodes. The generation stage in *MASOM* includes online machine listening. The agent can listen to itself and other performers by extracting eventfulness, pleasantness, and timbre features. *MASOM* uses machine listening module with the trained VMM to decide what to play next.

2.7 Evaluation of Musical Agents

Frayling [87] proposes three types of research in Art and Design. First, the research into art and design is the historical, aesthetic, and perceptual research such as the Music History research. Second, the research through art and design includes the research of materials, customization of technology, or procedures and results of practical experiments. Third, the research for art and design communicates the results of research through the end product, that is the work of art. The ideas and results are embodied in the artefact; hence, the verbal communication of the results is not the primary goal of this third type of research in art and design.

The developers of MuMe systems evaluate their implementations informally as a part of the software development. Hence, there are two classes of evaluation of MuMe systems: informal evaluations and formal evaluations. We differentiate the evaluation types of MuMe systems and musical agents with the following typology:

- **Informal Evaluations** does not involve formalized research methodologies.
 - **The authors** are the creators of MuMe systems.
 - **Users, peers and experts** are the close entourage of authors.
 - **The audience** is the recipients of the artworks generated by the MuMe systems.
 - **The media** covers critical writings of experts in art and music.
- **Formal Evaluations** are formalized methodologies to assess the success of MuMe systems.
 - **Peer reviewers, curators and jury** give direct and indirect feedback to the authors of MuMe systems.
 - **Theoretical and analytic measures** are the formal evaluation methodologies that does not involve human participants. These methodologies are synthetic measurements that are acknowledged in the academia.
 - **Empirical studies** apply quantitative, qualitative, and mixed methodologies with human participants.

2.7.1 Informal Evaluations

Informal evaluations do not involve any established research methodology. Informal evaluations of musical agents start at the beginning of system's ideation. The authors iterate the architecture and the system parameters as a part of the development process. This process includes many iterations in which the authors evaluate the system's output, change the parameters or the agent architecture, and evaluate the system's output again. For example, most machine learning algorithms require a set of parameters to be decided by the developer. These parameters are mostly set by many trials and errors with the system.

Colleagues and friends of authors are the subjects of another type of informal evaluations. The authors have a different perspective on the system with the feedback of people who are close to the authors. When the system's output is publicly shared, the authors receive feedback from the audience. Although this type of feedback is still informal, it is beneficial to evaluate the initial results of the system. When the system outputs reach the media such as journalists, critics, software testers, bloggers; the authors receive a feedback about social implications of systems.

2.7.2 Formal Evaluations

Formal evaluations use established research methodologies to answer a clearly defined research question to assess a system. Arges et al. [6] two types of formal evaluations: internal and external.

Internal formal evaluations are conducted during the generation stage of a musical agent. Musical agents assess their creative output during the generation to improve the output. In most cases, musical agent developers implement the internal evaluation as system feedback loops or using agents with evaluation roles. For example, Cypher (10) uses its listener agents to evaluate player agents. The internal evaluation is a part of system design, and we already covered the system design of musical agents in the previous sections.

External formal evaluations are conducted after the agent finishes its performance or generation. There are four aspects of external formal evaluations:

- **Dimensions of evaluation:** Three types of evaluation dimensions are common in the CC literature: software validation, the quality of a system's output, and creativity [162]. In MAS, the authors can study the creativity of one agent or the creativity of the system output. Most synthetic evaluations research the effect of hyper-parameters on the system output. A hyper-parameter is a common term in Machine Learning and it refers to the parameters of system design, such as number of agents in MAS, or genetic operator probabilities in EC. In most cases, hyper-parameters are set before the system run.
- **Participants of evaluation:** The authors develop MuMe systems to be used by a user to generate music that is presented to an audience. Therefore, researchers focus on three types of participants in the evaluation: the authors, the users, and the audience. Researchers can evaluate the research

and development process of authors, the interaction of users with the finalised system, and the audience response to the output generated by the system.

- **Output selection:** Ritchie [162] proposes five types of output selection that appears in CC: re-creating known exemplars of the domain, exploring the neighbourhood of these exemplars, exploring the parameter space of a system, random sampling of the parameter space, structured sampling of the parameter space. These selection options are also applicable to MuMe systems.
- **Methodology:** Software validations, synthetic evaluations, and empirical evaluations are the tools of formal evaluations.

Software Validations

The musical agents that we mention in this survey are written as software codes. Software evaluations use software validation techniques in Computer Science to assess if the code implementation is sound, complete, or stable. Briefly, the following three techniques come forward in Computer Science:

- **Formal Validations:** Mathematical proofs of the system behaviour are examples of this type of validations.
- **Black Box Tests:** Given pre-set inputs, black box tests study the system output to evaluate a system’s behaviour.
- **White Box Tests:** Given a set of selected inputs, white box tests exhaust a system for all possible conditions to ensure stability and robustness.

Synthetic Evaluations

Synthetic methods do not involve human participants. Theoretical, analytical, and computational tools are the methods of synthetic evaluations. Because of the particularities of musical agent implementations, the authors create new synthetic methodologies that are specific to their implementation. In the following, we survey synthetic evaluations of *Beatbender* (20), system (32), VMMAS (1), *IMAP* (36), and *pMIMACS* (46).

Levisohn and Pasquier [114] evaluated *Beatbender*’s (20) system output using two criteria: emergence and complexity. The authors assessed the emergent behaviours of the system by analyzing interaction between agents, and the complexity by comparing subsequent patterns generated by the system. The authors reported that *BeatBender* (22) successfully generated emergent rhythms by taking advantage of the Subsumption architecture.

Aucourtier [11] also evaluated the emergence and convergence in the multi-agent society (32) that evolves tuning systems. The author evaluated the system with different agent interaction types: single note shift interaction with harmonic timbre, drone shift interaction with harmonic timbre, drone shift interaction with compressed timbre, and drone shift interaction with a society of agents

with harmonic and compressed timbre. Hence, the author researched the effect of agent interaction types on the system output. Aucouturier concluded that the system could emerge coherent tuning systems through local agent interactions in the multi-agent society.

The following three evaluations studied the effect of hyper-parameters on systems' output. Vicari et al. [190] evaluated *VMMAS* ① with two evaluations. Both evaluations calculated a variable called *synchronism property*, which is calculated using the rhythm generated by the agents. However, the authors concealed the details of how to calculate this parameter. The authors claimed that synchronism property above 60 indicates a 'good performance'. The first evaluation included only software agents, and concluded that introducing new agents to *VMMAS* ① influenced the overall synchronism. Hence, the authors studied the effect of a hyper-parameter, that is the number of agents. We mention the second evaluation of *VMMAS* in Section Consensual Assessment Technique.

Miranda et al. conducted three evaluations to evaluate *IMAP*'s ③⑥ performance. The first evaluation studied if agents could perform according to their individual preferences. The weights of the rules in an agent's fitness function indicate the individual preferences of an agent. This evaluation concluded that average agent performances were correlated with the preferences of agents. The second evaluation showed that the user can control the overall diversity of performances by changing the spread of the rules weights. The third evaluation researched if the population was affected when a subset of agents in the population are biased in their fitness function. This third evaluation showed that *IMAP* ③⑥ could direct the performance diversity to a region in the search space by introducing a bias to a subset of the population.

Kirke and Miranda [108] analyzed *pMIMACS* ④⑥ output with a synthetic evaluation with three agents. This evaluation was the detailed analysis of two runs of MAS. The first run was 8 episode long whereas the second one was 10. During each episode, agents with performer roles performed for the agents with listener roles. For this evaluation, the agents were initiated with a unique melody including four notes. All notes were sixteenth notes generated by random walk. The authors claimed that *pMIMACS*'s outputs were less mechanical than the outputs that were 'usually produced by the algorithmic compositions systems.' However, the study did not include any empirical evaluation to support this claim.

None of the synthetic evaluations study the creativity of musical agents. However, musical agents tackle musical creative tasks, and the assessment of creativity is crucial to evaluate the success of a system.

Empirical Evaluations

Given that the definition of creativity is still in discussion [157], empirical evaluation methodologies handle the complexity of creativity assessment by using human participants to judge the output of a system. Before going into the details of these evaluations, let's cover the background of creativity.

Boden [29] defines creativity as 'the ability to generate new forms.' This definition explains creativity by focusing on the artifact. Boden continues by proposing *psychological* and *biological* creativity to categorize human and non-human creativity. *Biological* creativity is 'the ability to gen-

erate new cells, organs, organisms, or species.' We explored computational abstractions of biological creativity in musical agents in Section 2.5.2. In comparison, *psychological* creativity is 'the ability to generate ideas and/or artifacts that are new, surprising, and valuable.'

Boden focuses on two key points to understand which forms are new. First, Boden discusses the notion of *novelty* in creativity. Second, *historical creativity* is a special case of psychological creativity in which generated form is novel to the community. Furthermore, Boden states three types of creativity as a result of *mechanisms* generating novelty; *exploratory*, *combinational*, and *transformational*. First, *exploratory* creativity is making novel forms that satisfy constraints of a particular style. An example of exploratory creativity is improvising a Jazz melody in the style of Charlie Parker. Second, *combinational* creativity is combining styles in novel ways such as improvising a Jazz melody in the style of Chet Baker with the ornamentations of Charlie Parker. Third, *transformational* creativity is the expansion of known conceptual space. An example of transformational creativity is John Cage's idea of including random sounds of audience to a musical performance.

How to assess the creativity of a system has been a challenge for the CC research. Jordanous [106] pointed out the lack of evaluation in the publications of CC systems. Also, within the publications that evaluated their systems, the evaluation of creativity was not common. When the creativity evaluation took place, the participants were mostly the people who implemented the system. Jordanous emphasized the lack of evaluation criteria in the publications that evaluated creativity. According to Jordanous, there was a clear lack of connection between the evaluation of CC systems and the evaluation methodologies that were presented in CC. Currently, there is still no evaluation methodology that is accepted as a standard in CC. Jordanous also stated that CC inclined towards the evaluation of the quality in comparison to the evaluation of creativity. We observed similar tendencies in the evaluation of musical agents.

There has been recent attempts to categorise the evaluation methodologies for MuMe systems. Arges et al. [6] identified six types of external evaluations:

- Behavioral Tests
- Consensual Assessment Technique (CAT)
- Extensions within Computational Creativity
- Questionnaires, Correlational Studies, and Rating Scales
- Physiological measurements and neurophysiological measurements

Regarding the evaluation of creativity, we observed two common cases in the empirical evaluations of musical agents. In the first case, the authors evaluated the systems from the user perspective. The participants were expert users who tried the musical agent. Although these evaluations did not necessarily follow the typical CAT methodologies, we grouped them under the CAT category since the participants were expert users. In the second case, the authors evaluated a musical agent from the perspective of the audience. In these evaluations, the evaluation tools were questionnaires and

rating scales. Lastly, we observed only one system that incorporated an evaluation methodology from CC. In the following, we go into details of musical agent system evaluations on creativity.

Consensual Assessment Technique A group of experts evaluate the creativity of MuMe systems in Consensual Assessment Technique (CAT). In musical agents, the expert evaluation refers to quantitative and qualitative empirical evaluations with expert participants, and case studies. For example, the second evaluation of VMMAS ① [190] studied a performance session with software and human agents. The human performer reported that the system was successfully accompanied with satisfactory rhythmic and harmonic behavior.

Navarro et al. [145] presented an example of mixed method empirical evaluation study with *MUSIC-MAS* ⑥ that assists composers by generating harmony progressions. The participants were novice composers who were studying first year music theory at university. The authors asked the participants to compose their first piece using a harmony progression generated by *MUSIC-MAS*. The participants rated each others' compositions as well as *MUSIC-MAS*' success on assisted composition. This evaluation concluded that *MUSIC-MAS* could help novice composers by assisting composition tasks.

Murray-Rust and Smaill [140] carried out several case studies to evaluate their musical agent based on *MAMA* ⑨. The case studies were a duo of a human performer and

- another human performer
- a recording of a human performer
- a musical agent without expressivity and interactivity
- a musical agent mirroring the human input
- a musical agent including Musical Acts

However, this empirical evaluation concluded that the introduction of Musical Acts did not significantly change interactivity, competence and expressivity, and general performance.

Linson et al. [117] conducted two qualitative evaluations with *Odessa* ⑯ that is a mono-agent system including machine listening. The first study included eight expert musicians playing clarinet, trumpet, cello, soprano saxophone, guitar, bassoon, piano, and vocals. Six of eight musicians reported 'a process of familiarization and improved collaborative engagement' while two musicians were highly dissatisfied. After the first evaluation, the authors added a module of excitation to the architecture so that the system respond to higher activities in the input. The second qualitative evaluation had two expert musicians playing soprano saxophone and guitar. These musicians also participated in the first evaluation. The evaluation was a trio session including *Odessa*. The musicians reported a 'coherent identity' of *Odessa* regarding both evaluations.

Collins [50] evaluated *LL* ⑳ with two expert musicians. One of the experts was a percussionist whereas the other one was a violinist. Both sessions were presented as public concerts. The percus-

sionist conceptualized *LL* as the extensions of its programmer. The violinist mentioned the trade-off between the controllability versus agency of the system.

Similarly, Aucouturier and Pachet [12] pointed out the trade-off between autonomy and reactivity in the evaluation of *Ringomatic* (52). The evaluation was a case study of *Ringomatic*'s interaction with a human drum player. The authors clarified that *Ringomatic* could follow the human performer while preserving the global continuity.

Eigenfeldt and Pasquier [75] carried out a quantitative listening evaluation to evaluate *Coming Together:Freesound* (25). Four soundscape compositions were generated for the evaluation. One composition was generated by the system. Another one was generated by random. The remaining two were composed by an expert composer. One of the human composed ones was freely composed without constraints whereas the second one was limited with database, methods of processing, overall duration, static spatial distribution of four gestures in four channels. The evaluation survey questions focused on the soundscape characteristics, compositional success, skill level, and subjective reaction. In all cases, the system was better than the random generation.

Hawryshkewich et al. [101] carried out a case study with beginner drum players to test if *Beat-back* (51) could improve the self-directed learning of drum players. The authors reported that 'the majority of participants felt less enjoyment and more tension with drum zoning enabled.'

Surges and Dubnov [174] tested the capabilities of *PyOracle* (59) with a case study. *PyOracle* (59) was presented in a public concert as a performance with an expert musician. The case study was a *structured improvisation*. Structured improvisation is free improvisation with predefined constraints for musical sections. The case study was a concert performance including a score for both *PyOracle* and the human performer. The performance was followed up with an interview with the human performer. The performer emphasized that the flexibility of *PyOracle*'s timing mechanism could be elaborated.

Sampaio et al. [167] presented two empirical evaluations assessing the quality and diversity of *CinBalada*'s (64) musical output. The first evaluation showed that the participants preferred *CinBalada*'s output over random generated or similarity-based rhythms. The second evaluation concluded that the participants found the diversity of *CinBalada*'s output not too distant from the diversity of randomly generated rhythms.

In these evaluations, we observe that the details of methodology is not clear and the justification of the proposed methodology is missing. The main discussions around the formalization of expert studies is still to be done in MuMe. Notice that, the evaluation of *Odessa* (16), *LL* (25), and *PyOracle* (59) were conducted through post-performance interviews. These interviews did not follow a typical qualitative methodology. Regarding all CAT evaluations, the hypothesis is not clear and the dimensions of evaluation is vague. The evaluations are exploratory; however, this fact is implicit and there is no justification of why an exploratory approach is chosen.

Evaluation Methodologies of Computational Creativity This type of empirical evaluations integrate the methodologies of CC to evaluate MuMe systems. Many evaluation methodologies have

been proposed in CC, such as Standardised Procedure for Evaluating Creative Systems (SPECS) [106], the Creative Tripod [52], and FACE/IDEA model [156].

We have found only one study that incorporated a methodology from CC to evaluate musical agents. Yee-King and d'Inverno [203] used MusicCircle, a timeline-based tagging and annotation system to evaluate *SpeakeSystem* ⑥. The conclusion of the qualitative study was that the system gave a strong sense of interaction; however, failed to generate long-term structures.

By no means this survey covers all discussions around the assessment and evaluation of creativity and proposed evaluation methodologies in CC. Still, only one study incorporated a CC methodology to evaluate a musical agent. Hence, this creates opportunities to integrate CC evaluation frameworks to musical agents.

Questionnaires, Correlational Studies, and Rating Scales Surveys and questionnaires are one of the main tools that musical agent developers use to evaluate their applications. In comparison to CAT, the participant group is not a group of experts in this type of evaluations. We have found three systems with such evaluations.

Murray-Rust et al. conducted a questionnaire that is similar to the Turing Test [186] to evaluate their rhythm generating system *VirtuaLatin* ⑯. Turing test is one of the first methodologies that is proposed to evaluate automatic agents. Murray-Rust et al. concluded that the general public could not differentiate the machine-generated rhythm from a human-generated one while a higher percentage of expert listeners could.

Delgado et al. [58] evaluated *Inmamusys* ② by generating four compositions with the input affective labels *worry*, *happiness*, *chaos*, and *worry* again. The participants labelled these compositions with affective states of *sadness*, *happiness*, *fear*, *worry*, *chaos*, and *indifference*. The authors reported that the participants affective labels were in line with the input affect labels of the generated compositions.

Kirke and Miranda [109] conducted a listening test with ten participants to evaluate *MASC* ⑦. The evaluation aimed to see if the affective states of single agents could be observed in the melody output of the Multi-agent system. The results indicated that the affective states of single agents appeared in the final output. Still, the authors mentioned that a following this first evaluation with another one with more participants is required to conclude on a significant result.

2.7.3 Future Steps of Evaluation and Benchmarking

We observe that the evaluations of musical agents apply system specific methodologies. The dimensions of evaluations are not clear in most cases and the justification of why a particular dimension of a system is evaluated is missing. Given that the hypothesis and the methodologies of evaluations vary, no benchmarking tasks were initiated for musical agents. An obstacle for benchmarking is the reusability and code availability that we mention in Section 2.8.3.

The MIR field has developed a set of benchmarking tasks and through the Music Information Retrieval Evaluation eXchange (MIREX), the MIR field addresses formally defined challenges.

Musical Agents

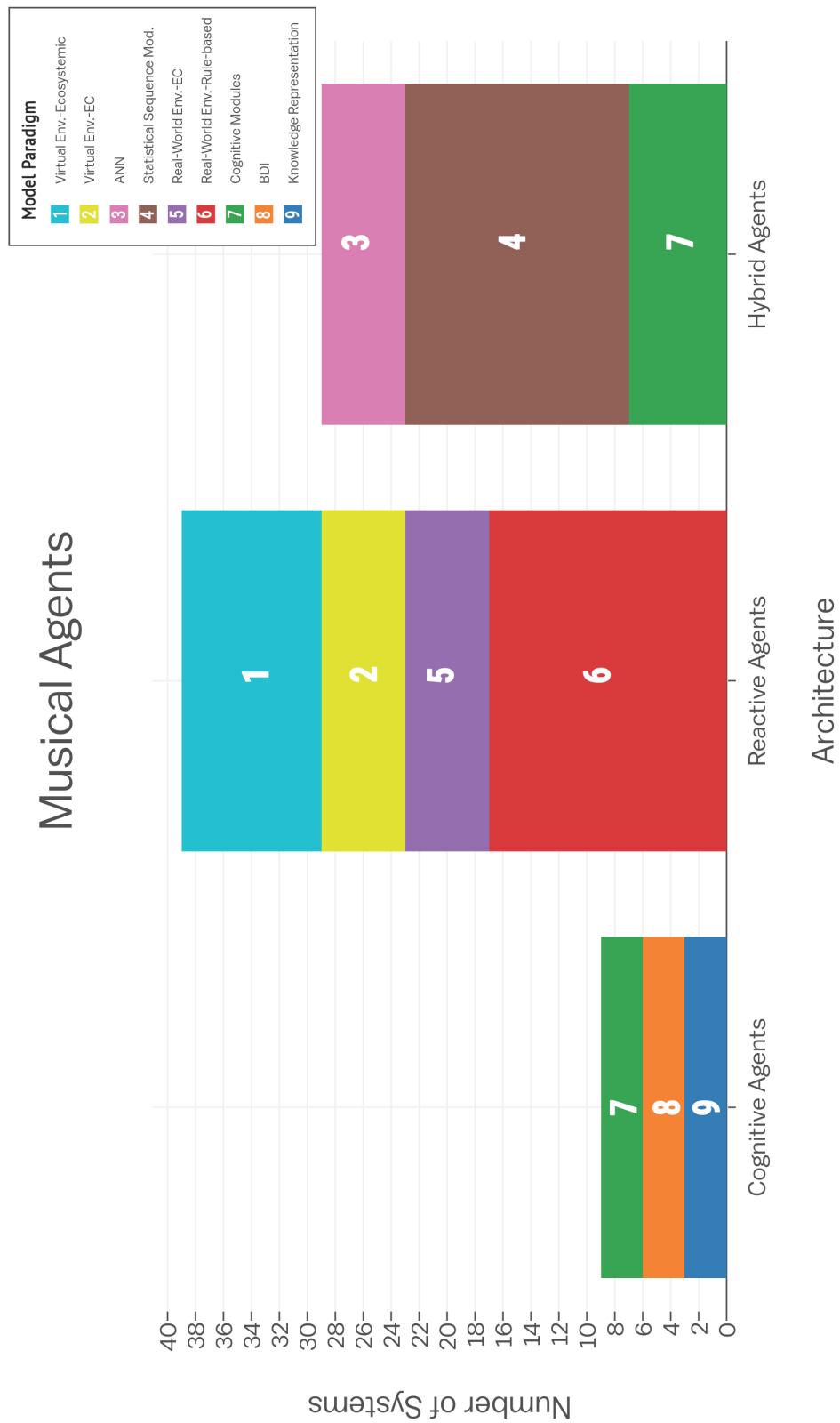


Figure 2.21: The number of musical agents per architecture type

Musical agent researchers could apply a similar approach for benchmarking. For example, many systems tackle style imitation tasks and it could be possible to benchmark these tasks. As of 2017, Institute Neukom have sent a call for Music Creative Turing Test 2018 ¹². This is a recent benchmarking attempt for MuMe systems including musical agents.

Although we covered the musical agent evaluations that applied consensual assessment technique, evaluation criteria from computational creativity, questionnaires, correlational studies, and rating scales; we found no study that applies behavioral tests, physiological and neurophysiological measurements. Behavioral tests assess divergent thinking, convergent thinking, artistic ability, and self assessment. Some examples of behavioral tests in Music are Measure of Musical Problem Solving [191] and Measure of Creative Thinking in Music II [194]. Physiological and neurophysiological measurements analyze the physiological response of audience. Motion capture, eye tracking, galvanic skin response, Electroencephalography (EEG) are the examples of tools to measure audience physiology. It could be also possible to apply the measurement neural responses of audience to evaluate the performance of a musical agent. However, these technologies are particular to specific areas and applications, and they are not always available in the institutes that research MuMe.

2.8 Ad Infinitum

2.8.1 Architectures and Algorithms

Figure 2.21 presents the number of systems for each architecture type. We observe that the number of reactive musical agents is the highest, followed by the number of hybrid musical agents. EC modules appear both in reactive agents in real-world and virtual environments. Moreover, cognitive modules show up in cognitive and hybrid agents. However, the number of cognitive musical agents is the lowest.

We have found only sixteen implementations of musical agents with cognitive modules, including the hybrid musical agents with cognitive models (Figure 2.21). Thórisson and Helgasson [184] present the state of the art cognitive architectures: *Ymir*, *ACT-R*, *Soar*, *NARS*, *OSCAR*, *AKIRA*, *CLARION*, *LIDA*, *Ikon Flux*. Except for *CLARION*, we have not encounter any study in which any of these architectures are applied to a MuMe task. Notice that, cognitive musical agent studies are challenging because applying a cognitive architecture to a musical task requires the expertise in Music, Computer Music, AI, MAS, and Cognitive Science. Also, these cognitive architectures are reasoning architectures and they are not music cognition architectures. The research on Music Perception and Cognition is still to be reflected to the cognitive musical agent studies.

Regarding musical agent studies with statistical sequence modelling, there is still more to be done to generate variety in longer musical sections. Pachet [153], and Assayag and Dubnov [8] clarify that Markov Models fail to represent the conditional probabilities of sequences longer than the order. Hence, many of the systems presented in Section 2.6.1 do not include long-term mem-

¹²<http://bregman.dartmouth.edu/turingtests/music2018>

ory and one can argue that these systems fail to produce variety in long-term musical sections and structures. Dubnov et al. [61] and Pachet [153] address this problem by introducing interactivity to Markov Models. Hence, the generation of long-term structures guided by a human performer. *Improtok* comes forward with the idea of using *scenario generation model*, and combining probabilistic methods with Factor Oracle is another promising approach to generate long-term continuity [65].

ANN algorithms are still to be examined by the musical agent developers. With the increasing research on Deep Learning, a variety of new algorithms as well as improvement of the previous algorithms are presented in the literature [5]. Briot et al. [39] surveyed Deep Learning approaches for musical tasks. Although these systems are mostly purely generative systems, it is possible to incorporate these approaches with MAS to develop musical agents. Moreover, we have found only one study (system 67 mentioned in Section 2.6.3) that evolves ANN modules using NeuroEvolution of Augmenting Topologies (NEAT) [32]. NEAT combines ANN with EC to evolve ANN modules.

Genetic Programming (GP) is a type of EC algorithms. We have not found any musical agents applying GP in the system design. GP, especially Cartesian Genetic Programming (CGP), has been applied to image recognition [99] as well as style imitation in Visual Arts [132]. Moreover, Wooldridge [201] proposes the idea of synthesizing agents. In all musical agent systems that we have covered, the authors develop the systems manually. Automatic musical agent design is possible using GP and CGP algorithms. GP and CGP have been applied to synthesize audio synthesis architectures [176, 195, 89, 2, 119]. It is possible to improve the automatic audio synthesizer design systems to synthesize musical agents.

2.8.2 Interdisciplinarity of MuMe

While the International Workshops on Musical Metacreation¹³ have covered MuMe topics for five MuMe workshops, five MuMe concerts, and three MuMe tutorials since 2012; the topics of MuMe field has been covered by various platforms such as International Computer Music Conference¹⁴, the International Symposium for Music Information Retrieval¹⁵, the Sound and Music Computing¹⁶, the Association for Computational Creativity¹⁷, the International Computer Music Association¹⁸, the International Conference on New Interfaces for Musical Expression¹⁹, Live Coding²⁰, the In-

¹³<http://musicalmetacreation.org/>

¹⁴<http://computermusic.org/>

¹⁵<http://www.ismir.net/>

¹⁶<http://smcnetwork.org/>

¹⁷<http://computationalcreativity.net/home/>

¹⁸<http://computermusic.org/>

¹⁹<http://www.nime.org/>

²⁰<https://toplap.org/>



Figure 2.22: The continuum of autonomy in musical agent design

ternational Symposium for the Electronics Arts²¹, the conferences held by the Association for the Advancement of Artificial Intelligence²².

In some cases, the success of MuMe systems are dependent on the advances in other disciplines. For example, many agents working with audio or hybrid I/O include machine listening. Examples of machine listening tasks are tempo estimation, fundamental pitch detection, sound similarity, rhythm similarity, melody similarity, affect estimation in sound, chord analysis, audio thumbnailing, novelty detection, etc. The MIR field addresses these tasks and many are still open research questions. By default, the success of musical agents with machine listening relies on the quality of the machine listening algorithm. Hence, these musical agents are dependent on the advances in MIR studies.

Therefore, the MIR and MuMe fields naturally benefit from each other by putting forward new problems and solutions. For example, MuMe uses advanced technologies of MIR in machine listening and automatic extraction of higher level music features. Also, musical agents can utilize the recent developments in Affective Computing in Sound and Music [66, 82] in machine listening modules of musical agents. Other MIR areas are also valuable to musical agents such as musically informed audio decomposition, tempo and beat tracking, chord recognition, and music structure analysis [138]. Likewise, MIR can take advantage of the MuMe research. For example, Collins [51] proposed autonomous critic agents in the assessment of musical style, novelty, or quality. Collins's study proposed a model for critic agents that have listened to more music than humans could. Such critic agents can be explored for the tasks of recommendation systems in MIR.

2.8.3 Design Considerations

The developers of musical agents create a system architecture by going through a design process. Autonomy in musical agent design ranges from encoded systems to agent synthesis. The developers design the system architecture manually in the encoded and heuristics systems. In comparison, agent synthesis is completely autonomous [201] and can generate musical agent architectures. We propose to refer to the phenomena of agent synthesis as Metacreation of Metacreation (Meta²creation). We

²¹<http://www.isea-web.org/>

²²<https://www.aaai.org/>

claim that Musical Meta²creation is developing systems that create systems that partially or completely automatize musical tasks.

Machine Learning lies in the middle of the autonomy continuum in the musical agent design. The developers of musical agents often incorporate Machine Learning in their system design. Machine Learning algorithms have parameters to be set by the developers. For example, an EC algorithm has genetic operator probabilities that set the chance of applying the genetic operators to an individual. Another example is the highest order parameter in Variable Markov Models. The developers often set these parameters by listening to the system output for various parameter options. This process is addressed in the Machine Learning as Interactive Machine Learning or User-Centered Machine Learning [14, 93]. The research on the procedures, tendencies, and underlying factors of developing musical agents is still to be done.

We have encountered three recent studies that studied the design principles of Computational Creativity (CC) systems including musical agent systems [35, 36, 37]. First, Bray and Bown [35] compare user experience of a DAW and the musical agent Nodal (42). Second, Bray and Bown [36] propose applying the Interaction Design theory to CC systems. Third, Bray et al. [37] compared three generative music systems to understand the effect of the degree of encapsulation in MuMe systems. The study included a direct manipulation system, a programmable interface system, and a highly encapsulated system.

Reusability and code availability is another issue of musical agents. Out of 78 systems, the source codes of 18 systems (23%) are available to the public (Table 2.1). This makes the comparison of different musical agents difficult. Addressing this issue, the manifesto of Musebot framework encourages making musical agents open-source by publicly sharing the code of the framework and submitted musical agents [34]. *Musebot* project is a framework for musical agents which allows interactive live performances with human performers and multiple musical agents [79, 73]. The system design of the framework is the client/server architecture in MAS. As of 2017, Musebot framework is compatible with MAX, Max for Live, PureData, Processing, SuperCollider, Python, Extempore, and JAVA. The framework provides exciting opportunities such as collaborative performances of various musical agents and autonomous curation of musical agent ensembles. The Musebot framework provides an opportunity to create a public repository of musical agents.

Table 2.1 shows which systems have been presented to the public within our knowledge. We also include systems implementing assisted composition tasks, if the systems have been used to produce a composition that is presented to the public. Out of 78 system, 39 systems (50%) have been presented in public venues. Musical agents can aim for increasing the percentage of systems available to the public.

2.8.4 MuMefication

MuMe as a Field

There is an objective evidence that MuMe is an interdisciplinary field. In a recent paper devoted to this topic, Bodily and Ventura [30] clarify that total 80 papers were published in the five MuMe Workshops between 2012 and 2017. These papers had a total 111 authors. Out of these 111 authors, 88 (79.2%) published only once, 13 (11.7%) published twice, and 8 (7.2%) published three or more times in MuMe Workshops. Out of these 80 papers, 36 of them had 173 external citations in total. In comparison, there were only 13 instances where MuMe papers cited other MuMe papers. The higher rate of external citations in comparison to internal citations of MuMe publications, and low re-publication rate of authors indicate that MuMe is a growing interdisciplinary field. [30] mention that the external citations of MuMe papers appeared in papers presented in a variety of venues such as the International Conference on Computational Creativity (ICCC); Computers in Entertainment (CIE); the Computer Music Journal (CMJ); the Sound and Music Computing Conference (SMC); and the International Computer Music Conference (ICMC).

We propose that MuMe as an interdisciplinary field inherits both practice of generative music whether it is artistic, heuristic, creative AI; as well as the scientific study of Computational Creativity for musical creative tasks. We think that the term MuMe can provide a formalization of such systems using the definitions and ideas of Computational Creativity, Generative Art, and Artificial Intelligence. We propose that MuMe as an interdisciplinary field aims to bring together all fields that apply autonomous approaches for creative musical tasks. Our definition of MuMe as a field is inclusive in the sense that it is not creative musical AI because it is not always AI, it is not a simulation of musical creativity because MuMe can also cover *creativity as it could be*, it is not necessarily live-coding because MuMe systems are not necessarily performed live, it is not artificial life because it also covers systems that do not simulate a virtual environment. In addition, we think that CC literature gives an established ground to explain whether MuMe systems are musically creative, and if they are, the kind of creativity that MuMe systems output.

A Typology of MuMe Systems

The discussion around the typology of MuMe systems is still ongoing. Eigenfeldt et al. [78] propose a taxonomy of MuMe systems in seven levels of independence, compositionality, generativity, proactivity, adaptability, versatility, and volition; ordered from least autonomous to the most. Based on these seven levels of MuMe systems, we propose six levels of musical agents which are ordered from the lowest level to the highest one:

1. Reactivity: Agents respond to the changes in the environment in a timely fashion.
2. Proactivity: Agents can perceive their environment and plan future actions.
3. Interactivity: Agent can interact with other agents (human, artificial, or biological).

4. Adaptability: Agents learn from their environment to improve competence or efficiency.
5. Versatility: Agents are domain independent.
6. Volition and framing: Agents can explain why they choose certain actions when asked by other agents.

The higher levels can inherit properties of the lower levels whereas the lower levels cannot present the distinctive properties of the higher levels. Many agents that we cover demonstrate reactivity behaviours. For example, *Odessa* (16) can react to the musical actions of other performers. *Odessa* also exhibits interactivity by influencing other agents by actions. *Odessa* diverges from the current state of the environment if other agents fail to generate variability. However, *Odessa* does not learn from the environment. In comparison, system (69) exhibits adaptivity by training the SOM in the architecture online. The musical agents that are free of the author's style or choice show behaviours of Versatility. For example, the *Continuator* (50) is a flexible agent that imitates the style of any performer.

Although the author's style is not explicitly embedded in the *Continuator*, one can argue that by just choosing one generative algorithm over the other, the authors make implicit stylistic choices on the design of the musical agent. Thomas et al. [183] studied the bias of three style imitation algorithms in melody generation. The authors compared VMM, FO, and *MusiCOG* (8) and concluded that each algorithm introduced a particular bias to the melody generation. One can argue that the *Continuator* is not completely independent of the author's style because the author made the decision on the generative algorithm that was used in the system design. Hence, the selection of one melody generation algorithm rather than others introduced a particular bias to the *Continuator*.

The taxonomy of Eigenfeldt et al. [78] distinguish MuMe systems based on the dimension of autonomy. In comparison, Blackwell et al. [26] propose a taxonomy of MuMe systems by focusing on the system architecture. The authors propose a new term: live algorithms. Live algorithms include musical agents as well as purely generative music systems that do not utilize any input in the system design. The authors clarify four main interaction types of Live Algorithms: autonomy, novelty, participation, and leadership. The authors present eight case scenarios of system designs. Each case has a different combination of incoming audio stream, outgoing audio stream, human control, and three modules of P (listening/analysis), Q (performing/synthesis) and f (patterning, reasoning or intuition). Moreover, the authors state four types of Live Algorithm behaviours: shadowing, mirroring, coupling and negotiation. The authors continue by presenting implementations of Live Algorithms and further considerations.

2.8.5 Challenges and Opportunities

There are several reasons why the research and development of generative systems and musical agents matter. The main usage of computational systems have shifted from rational problem solving.

With the increasing number of personal computational systems, the percentage of computational power that is used for entertainment, art, and culture increased rapidly.

As a result, the demand for generative systems including musical agents in the creative industries escalated. This demand arises from the growth of non-linear media. Non-linear media enable users to choose from the available options in the media. Hence, non-linear media are interactive by nature. Two examples of non-linear media are games and websites. The workload to generate content for non-linear media is vastly greater than the workload of linear media production. Hence, there is an increasing demand for autonomous, adaptive systems that can fulfil the requirements of non-linear content. In that sense, we can use musical agents in the industry of non-linear media as adaptive and autonomous systems making music. Moreover, these autonomous systems can enable the personalization of the content. That is, the software can adapt to the user's specific choices, aesthetics, and requirements.

Multi-agent systems are applied to simulate real-world phenomena. We can apply musical agents to simulate and study musical phenomena. Using such simulations, we can model and study (software or human) agent interactions and emergent behaviours in musical tasks. In MuMe, this is referred as modelling creativity as it is [155]. These simulations can help understanding how we make music.

In comparison to musical creativity as it is, musical agents introduce new opportunities for the exploration of musical creativity as it could be. One advantage of software agents is that agents can both play music, listen, and exchange messages about their beliefs, desires, and intentions during a performance. The rate of communication can be much higher than that of human communication. Also, software agents can be shared easily over the internet and this creates new collaboration opportunities that go beyond logistic restrictions such as the location and attendance of performers.

Wiggins [199] formalizes Boden's definition of creativity with a framework called Creative Systems Framework (CSF). CSF also includes a conceptual space, a rule set that defines the conceptual space, a rule set that defines how agents can traverse the space, an evaluation rule set that assesses the value and novelty of concepts. Wiggins concludes that exploratory creativity at the meta-level is, in fact, transformational creativity. Hence, Wiggins emphasizes search approaches in Computational Creativity studies.

VMO, *FILTER* and *MASOM* are three systems that apply the idea of search in the conceptual space. These systems define a conceptual, multi-dimensional musical space. The dimensions of the space are audio features, i.e. sound properties. *MASOM* applies VMM, and *FILTER* implements FO for statistical sequence modelling. *VMO* is a model that combines VMM with FO. Following their work on VOM, [193] Wang and Dubnov combine HMM and VMO for the MuMe task of harmony generation. Although this work applies style imitation with symbolic representation of music, Wang and Dubnov compare VMO with HMM-GMM and K-Means machine learning algorithms. Wang and Dubnov conclude that in the conceptual feature space, VMO models temporal relationships whereas HMM-GMM and K-Means clusters spatially.

VMO, *FILTER* and *MASOM* define the musical form as a traversal in this multi-dimensional space. Since we can mathematically model a traversal in a multi-dimensional space, two Metacreative opportunities arise: style combination and style transformation. Style combination is combining different attributions of styles to come up with a new style. This corresponds to combinational creativity in Boden’s taxonomy of creativity. However, we do not know if the process of combining styles is linear in mathematical forms. If we combine two styles, do we explore a region that is an intersection for these two styles? A generalized version of this question is, how do we traverse the conceptual musical space by combining different attributions of two styles? Style transformation is applying a transformation function to a style to come up with another style or a new style. If we can model musical form and musical style mathematically, we can also define a function that transforms one style to another. We can also explore variety of transformation functions to research new styles.

The algorithms that we cover in this survey introduce biases and some of these biases have been pointed out in the literature. Thomas et al. [183] compares Markov Models, FO, and *MusiCOG* ⑧ on the task of melody generation and concluded that Markov Models and FO deviated from the training corpus. This indicates that the authors of Metacreative systems may introduce biases to the creative output of their systems by choosing one algorithm over others. Further research is required to clarify what kind of transformations machine learning algorithms introduce to musical agents as well as MuMe systems. Then, we could approach these algorithms as transformation functions that we apply on musical creative tasks.

2.9 Conclusion

Autonomous computational systems have been applied to various musical tasks. Multi-agent systems and Artificial Intelligence technologies exemplify autonomy in computational systems. Musical agents utilize artificial agent architectures and Multi-agent systems to automatize musical tasks of composition, assisted composition, interpretation, improvisation, accompaniment, melody, rhythm, and harmony generation, continuation, style imitation, arrangement, curation. We surveyed 78 musical agents systems whose architectures have been presented in peer-reviewed platforms. We proposed a typology of musical agents that is framed around the terminologies of Generative Music, Computational Creativity, Artificial Intelligence, Metacreation, and Musical Metacreation fields. This typology presents musical agents in 9 dimensions of agent architectures, musical tasks, environment types, number of agents, number of agent roles, communication types, corpus types, input/output types and human interaction modality. Our survey has given the details of musical agents by grouping the systems according to the architecture types. We incorporated the architecture types of cognitive, reactive, and hybrid architectures in Multi-agent systems to classify musical agents. We further categorized musical agents using the architecture model paradigms. As a special case, we also used environment types to further group the reactive musical agents. Within each section, we grouped musical agents by the musical task that they carry out. We hope that this organization of

the survey guides the reader to have an understanding of what has been done in the interdisciplinary field of musical agents.

We mentioned in Section 2.2 that creative tasks lack quality measures, which highlights the difficulties of evaluating musical agents. We suggested a classification of evaluation of musical agents where the specific evaluation methodologies of systems that we survey is incorporated to the corresponding classes. We ended this section by highlighting a possibility of benchmarking tasks for musical agents. We started our final section with an overlook of architectures and algorithms that have been covered by musical agents, which indicate the opportunities of different architecture types that can be applied to musical agents. Towards our conclusion, we introduce Musical Metacreation as a field, mention the interdisciplinarity of the field and design consideration of MuMe systems. We proposed six levels of musical agents, which is a derivation of seven levels of MuMe systems proposed in the literature. Before conclusion, we remark the challenges and opportunities in musical agents, and indicate several reasons why the research of musical agents as well as MuMe matters.

The studies of MuMe field aim to guide musicians and artists to understand musical creativity and find new ways of musical creativity. We hope that this review of musical agents helps both researchers and practitioners to understand and design autonomous software making music. Almost all studies mentioned in this review are presented in the last two decades. With the increasing research on MAS and AI, we are confident that musical agents will influence and contribute to how we make music in the future.

Bibliography

- [1] Asmaa Majid Al-Rifaie and Mohammad Majid Al-Rifaie. Generative Music with Stochastic Diffusion Search. In Colin Johnson, Adrian Carballal, and João Correia, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, number 9027 in Lecture Notes in Computer Science, pages 1–14. Springer International Publishing, April 2015. ISBN 978-3-319-16497-7 978-3-319-16498-4. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-319-16498-4_1.
- [2] Alo Allik. Gene expression synthesis. In *the Proceedings of the Joint Conference ICMC14-SMC14*, Athens, Greece, 2014. URL http://tehis.net/acquire/AloAllik_GeneExpressionSynthesis.pdf.
- [3] Alvaro Ámorim, Luís Fabrício W. Góes, Alysson Ribeiro da Silva, and Celso França. Creative Flavor Pairing: Using RDC Metric to Generate and Assess Ingredients Combinations. In *Proceedings of the Eight International Conference on Computational Creativity (ICCC 2017)*, 2017. URL http://computationalcreativity.net/iccc2017/ICCC_17_accepted_submissions/ICCC-17_paper_40.pdf.

- [4] Daichi Ando and Hitoshi Iba. Real-time Musical Interaction between Musician and Multi-agent System. In *Proceedings of the 8th Generative Art Conference*, 2005. URL <http://www.iba.t.u-tokyo.ac.jp/papers/2005/dandoGA2005.pdf>.
- [5] I Arel, D C Rose, and T P Karnowski. Deep Machine Learning - A New Frontier in Artificial Intelligence Research. *IEEE Computational Intelligence Magazine*, 5(4):13–18, November 2010. ISSN 1556-603X. doi: 10.1109/MCI.2010.938364. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5605630>.
- [6] Kat Arges, Jamie Forth, and Geraint A. Wiggins. Evaluation of musical creativity and musical metacreation systems. *Computers in Entertainment (CIE) - Special Issue on Musical Metacreation, Part II*, 14(3), 2016. URL <https://qmro.qmul.ac.uk/xmlui/handle/123456789/14230>.
- [7] Jaime Arias, Myriam Desainte-Catherine, and Shlomo Dubnov. Automatic Construction of Interactive Machine Improvisation Scenarios from Audio Recordings. In *The Fourth International Workshop on Musical Metacreation (MUME 2016)*, 2016. URL <https://hal.archives-ouvertes.fr/hal-01336825/>.
- [8] G. Assayag and S. Dubnov. Using Factor Oracles for Machine Improvisation. *Soft Computing*, 8(9):604–610, August 2004. ISSN 1432-7643, 1433-7479. doi: 10.1007/s00500-004-0385-4. URL <http://link.springer.com.proxy.lib.sfu.ca/article/10.1007/s00500-004-0385-4>.
- [9] Gérard Assayag, Shlomo Dubnov, and Olivier Delerue. Guessing the Composer’s Mind: Applying Universal Prediction to Musical Style. In *Proceedings of the 1999 International Computer Music Conference, ICMC 1999*, page 6, Beijing, China, 1999.
- [10] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. Omax brothers: a dynamic topology of agents for improvisation learning. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 125–132. ACM Press, 2006. URL <http://dl.acm.org/citation.cfm?id=1178742>.
- [11] Jean-Julien Aucouturier. Artificial Evolution of Tuning Systems. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.
- [12] Jean-Julien Aucouturier and François Pachet. Ringomatic: A Real-Time Interactive Drummer Using Constraint-Satisfaction and Drum Sound Descriptors. In *In Proceedings of the International Conference on Music Information Retrieval*, pages 412–419, 2005. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.4681&rep=rep1&type=pdf>.

- [13] Pascal Baltazar, De La Hogue Théo, and Myriam Desainte-Catherine. Demo: i-score, an Interactive Sequencer for the Intermedia Arts. 2014. doi: 10.13140/2.1.3556.0008.
- [14] Francisco Bernardo, Michael Zbyszynski, Rebecca Fiebrink, and Mick Grierson. Interactive Machine Learning for End-User Innovation. In *Proceedings of AAAI Spring Symposium*. American Association for Artificial Intelligence (AAAI), 2016. URL <http://research.gold.ac.uk/id/eprint/19767>.
- [15] Frédéric Bevilacqua, Bruno Zamborlin, Anthony Sypniewski, Norbert Schnell, Fabrice Guédy, and Nicolas Rasamimanana. Continuous realtime gesture following and recognition. In *International gesture workshop*, pages 73–84. Springer, 2009.
- [16] Peter Beyls. Interaction and Self-organisation in a Society of Musical Agents. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (Musical 2007)*, 2007. URL http://cmr.soc.plymouth.ac.uk/publications/musical_beyls.pdf.
- [17] Peter Beyls. On-line Development of Man-Machine Relationships: Motivation-driven Musical Interaction. In *Proceedings of the 11th Generative Art Conference*, Milan, Italy, 2008. URL <http://www.generativeart.com/on/cic/papersGA2008/1.pdf>.
- [18] Peter Beyls. Interactive Composing as the Expressions of Autonomous Machine Motivations. In *Proceedings of the International Computer Music Conference (ICMC 2009)*, Montreal, Canada, 2009.
- [19] Peter Beyls. Structural Coupling in a Society of Musical Agents. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.
- [20] Peter Beyls. Autonomy, Influence and Emergence in an Audiovisual Ecosystem. In *Proceedings of the Generative Arts Conference, Rome, Italy*, 2012. URL <http://www.generativeart.it/GA2012/peter.pdf>.
- [21] Peter Beyls, Gilberto Bernardes, and Marcelo Caetano. earGram Actors: An Interactive Audiovisual System Based on Social Behavior. *Journal of Science and Technology of the Arts*, 7(1):43–54, 2015. URL <http://artes.ucp.pt/citarj/article/download/142/104>.
- [22] John Biles. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–131. International Computer Music Association, 1994. URL <http://igm.rit.edu/~jabics/GenJam94/Paper.html>.
- [23] John A. Biles. Performing with Technology: Lessons Learned from the GenJam Project. In *Proceedings of the 2nd International Workshop on Musical Metacreation (MUME 2013)*, Boston, MA, 2013. URL <http://musicalmetacreation.org/>

mume2013/content/proceedings/Performing%20with%20Technology-%20Lessons%20Learned%20from%20the%20GenJam%20Project.pdf.

- [24] Jim Bizzocchi, Arne Eigenfeldt, and Miles Thorogood. Generating Affect: Applying Valence and Arousal values to unified video, music, and sound generation system. In *Proceedings of the 18th Generative Art Conference*, volume 49, pages 621–630, Venice, 2015. URL http://www.generativeart.com/ga2015_WEB/generating-affect_eigenfeldt.pdf.
- [25] Tim Blackwell and Michael Young. Self-organised music. *Organised Sound*, 9(02):123–136, 2004. URL http://journals.cambridge.org/abstract_S1355771804000214.
- [26] Tim Blackwell, Oliver Bown, and Michael Young. Live Algorithms: Towards Autonomous Computer Improvisers. In Jon McCormack and Mark d’Inverno, editors, *Computers and Creativity*, pages 147–174. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-31726-2 978-3-642-31727-9. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-642-31727-9_6.
- [27] Georges Bloch, Shlomo Dubnov, and Gérard Assayag. Introducing video features and spectral descriptors in the omax improvisation system. In *International Computer Music Conference ’08*, 2008. URL <https://hal.archives-ouvertes.fr/hal-01161405/>.
- [28] Margaret A Boden. Computer models of creativity. *AI Magazine*, 30(3):23, 2009.
- [29] Margaret A. Boden. Creativity and ALife. *Artificial Life*, 21(3):354–365, August 2015. ISSN 1064-5462, 1530-9185. doi: 10.1162/ARTL_a_00176. URL http://www.mitpressjournals.org/doi/10.1162/ARTL_a_00176.
- [30] Paul M Bodily and Dan Ventura. Musical Metacreation: Past, Present, and Future. In *Proceedings of the Sixth International Workshop on Musical Metacreation*, page 5, 2018.
- [31] O. Bown, J. McCormack, and T. Kowaliw. Ecosystemic methods for creative domains: Niche construction and boundary formation. In *2011 IEEE Symposium on Artificial Life (ALIFE)*, pages 132–139, April 2011. doi: 10.1109/ALIFE.2011.5954651.
- [32] Oliver Bown. Experiments in Modular Design for the Creative Composition of Live Algorithms. *Computer Music Journal*, 35(3):73–85, 2011. ISSN 1531-5169. URL https://muse-jhu-edu.proxy.lib.sfu.ca/journals/computer_music_journal/v035/35.3.bown.html.
- [33] Oliver Bown and Aengus Martin. Backgammon: Process-based Musical Explorations Using the Agent Designer. In *Proceedings of the 9th ACM Conference on Creativity & Cognition, C&C ’13*, pages 390–391, New York, NY, USA, 2013. ACM Press. ISBN 978-

- 1-4503-2150-1. doi: 10.1145/2466627.2481236. URL <http://doi.acm.org/10.1145/2466627.2481236>.
- [34] Oliver Bown, Benjamin Carey, and Arne Eigenfeldt. Manifesto for a Musebot Ensemble: A Platform for Live Interactive Performance Between Multiple Autonomous Musical Agents. In *Proceedings of the International Symposium of Electronic Art 2015 (ISEA 2015)*, 2015. URL http://isea2015.org/proceeding/submissions/ISEA2015_submission_141.pdf.
 - [35] Liam Bray and Oliver Bown. Linear and non-linear composition systems: User experience in nodal and pro tools. In *Proceedings of the Australian Computer Music Association Conference*, 2014. URL <http://www.liambray.info/s/Linear-and-Non-linear-Composition-Systems-7dnx.pdf>.
 - [36] Liam Bray and Oliver Bown. Applying Core Interaction Design Principles to Computational Creativity. In *Proceedings of the Seventh International Conference on Computational Creativity*, 2016. URL <http://www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/Applying-Core-Interaction-Design-Principles-to-Computational-Creativity.pdf>.
 - [37] Liam Bray, Oliver Bown, and Benjamin Carey. How Can We Deal With The Design Principle Of Visibility In Highly Encapsulated Computationally Creative Systems? In *Proceedings of the Eighth International Conference on Computational Creativity*, Atlanta, Georgia, USA, 2017.
 - [38] Mason Bretan and Gil Weinberg. A survey of robotic musicianship. *Communications of the ACM*, 59(5):100–109, April 2016. ISSN 00010782. doi: 10.1145/2818994. URL <http://dl.acm.org/citation.cfm?doid=2930840.2818994>.
 - [39] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep Learning Techniques for Music Generation-A Survey. *arXiv preprint arXiv:1709.01620*, 2017.
 - [40] Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1087032.
 - [41] Rodney A. Brooks. Intelligence without reason. In *The artificial life route to artificial intelligence: Building embodied, situated agents*, pages 25–81. L. Erlbaum Associates Inc., NJ, USA, 1995. URL <http://idlebrain.yolasite.com/resources/Article%20-%20AI.pdf>.
 - [42] Joanna Bryson. The Reactive Accompanist: Adaptation and Behavior Decomposition in a Music System. In Luc Steels, editor, *The Biology and Technology of Intelligent Autonomous*

- Agents*, pages 365–376. Springer, Berlin, Heidelberg, 1995. ISBN 978-3-642-79631-9 978-3-642-79629-6. URL http://www.springerlink.com/index/10.1007/978-3-642-79629-6_15.
- [43] Bruce G Buchanan. Creativity at the Metalevel AAAI-2000 Presidential Address. *AI Magazine*, 22(3):16, 2001.
- [44] Antonio Camurri, Alessandro Catorcini, Carlo Innocenti, and Alberto Massari. Music and Multimedia Knowledge Representation and Reasoning: The HARP System. *Computer Music Journal*, 19(2):34, 1995. ISSN 01489267. doi: 10.2307/3680599. URL <http://www.jstor.org/stable/3680599?origin=crossref>.
- [45] Pietro Casella and Ana Paiva. Magenta: An architecture for real time automatic composition of background music. In *Intelligent Virtual Agents*, pages 224–232. Springer, 2001. URL http://link.springer.com/chapter/10.1007/3-540-44812-8_18.
- [46] Justine Cassell, Joseph Sullivan, Elizabeth Churchill, and Scott Prevost. *Embodied Conversational Agents*. MIT Press, 2000. ISBN 978-0-262-03278-0. Google-Books-ID: tHiKZGh9t7sC.
- [47] Nick Collins. Drumtrack: Beat induction from an acoustic drum kit with synchronised scheduling. In *Proceedings of International Computer Music Conference (ICMC)*, 2005. URL <http://community.dur.ac.uk/nick.collins/research/drumtrack.pdf>.
- [48] Nick Collins. BBCut2: Integrating beat tracking and on-the-fly event analysis. *Journal of New Music Research*, 35(1):63–70, March 2006. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298210600696600. URL <http://www.tandfonline.com/doi/abs/10.1080/09298210600696600>.
- [49] Nick Collins. Reinforcement learning for live musical agents. In *Proceedings of the International Computer Music Conference (ICMC), Belfast*, 2008. URL <http://users.sussex.ac.uk/~nc81/research/rlforlivemusicalagents.pdf>.
- [50] Nick Collins. LL: Listening and learning in an interactive improvisation system. Technical report, University of Sussex, 2011.
- [51] Nick Collins. Towards Machine Musicians Who Have Listened to More Music Than Us: Audio Database-Led Algorithmic Criticism for Automatic Composition and Live Concert Systems. *Computers in Entertainment*, 14(3):1–14, January 2017. ISSN 15443574. doi: 10.1145/2967510. URL <http://dl.acm.org/citation.cfm?doid=3023312.2967510>.

- [52] Simon Colton. Creativity Versus the Perception of Creativity in Computational Systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, volume 8, 2008.
- [53] Simon Colton and Geraint A. Wiggins. Computational Creativity: The Final Frontier? *Frontiers in Artificial Intelligence and Applications*, pages 21–26, 2012. ISSN 0922-6389. doi: 10.3233/978-1-61499-098-7-21. URL <http://www.medra.org/servlet/aliasResolver?alias=iospressISSNISBN&issn=0922-6389&volume=242&spage=21>.
- [54] Darrell Conklin. Multiple Viewpoint Systems for Music Classification. *Journal of New Music Research*, 42(1):19–26, March 2013. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2013.776611. URL <http://www.tandfonline.com/doi/abs/10.1080/09298215.2013.776611>.
- [55] Arshia Cont, Shlomo Dubnov, and Gérard Assayag. Anticipatory model of musical style imitation using collaborative and competitive reinforcement learning. In *Anticipatory behavior in adaptive learning systems*, pages 285–306. Springer, 2007. URL http://link.springer.com/chapter/10.1007/978-3-540-74262-3_16.
- [56] Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Perennial Modern Classics, New York, 1 edition edition, July 2008. ISBN 978-0-06-133920-2.
- [57] Palle Dahlstedt and Mats G. Nordahl. Living melodies: Coevolution of sonic communication. *Leonardo*, 34(3):243–248, 2001. URL <http://www.mitpressjournals.org/doi/pdf/10.1162/002409401750287010>.
- [58] Miguel Delgado, Waldo Fajardo, and Miguel Molina-Solana. Inmamusys: Intelligent Multi-agent Music System. *Expert Systems with Applications*, 36(3):4574–4580, April 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2008.05.028. URL <http://dx.doi.org/10.1016/j.eswa.2008.05.028>.
- [59] Alexandre Donze, Rafael Valle, Ilge Akkaya, Sophie Libkind, Sanjit A Seshia, and David Wessel. Machine Improvisation with Formal Specifications. In *the Proceedings of the Joint Conference ICMC14-SMC14*, page 8, Athens, Greece, 2014.
- [60] Shlomo Dubnov and Gérard Assayag. Improvisation Planning and Jam Session Design using concepts of Sequence Variation and Flow Experience. In *Proceedings of Sound and Music Computing 2005*, page 7, Italy, 2005.
- [61] Shlomo Dubnov, Gérard Assayag, and Ran El-Yaniv. Universal Classification Applied to Musical Sequences. In *Proceedings of the 1998 International Computer Music Conference, ICMC 1998*, Ann Arbor, Michigan, USA, 1998.

- [62] Shlomo Dubnov, Stephen McAdams, and Roger Reynolds. Structural and affective aspects of music from statistical audio signal analysis. *Journal of the American Society for Information Science and Technology*, 57(11):1526–1536, September 2006. ISSN 15322882, 15322890. doi: 10.1002/asi.20429. URL <http://doi.wiley.com/10.1002/asi.20429>.
- [63] Shlomo Dubnov, Gerard Assayag, and Arshia Cont. Audio Oracle: A New Algorithm for Fast Learning of Audio Structures. In *Proceedings of International Computer Music Conference*, 2007. URL <https://hal.inria.fr/hal-00839072/document>.
- [64] Shlomo Dubnov, G. Assayag, and A. Cont. Audio Oracle Analysis of Musical Information Rate. In *Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC)*, pages 567–571, September 2011. doi: 10.1109/ICSC.2011.106.
- [65] Ken Déguelnel, Emmanuel Vincent, and Gérard Assayag. Probabilistic Factor Oracles for Multidimensional Machine Improvisation. *Computer Music Journal*, 42(2):52–66, 2018.
- [66] Tuomas Eerola and Jonna K. Vuoskoski. A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli. *Music Perception: An Interdisciplinary Journal*, 30(3):307–340, February 2013. ISSN 07307829, 15338312. doi: 10.1525/mp.2012.30.3.307. URL <http://mp.ucpress.edu/cgi/doi/10.1525/mp.2012.30.3.307>.
- [67] Arne Eigenfeldt. Emergent Rhythms through Multi-agency in Max/MSP. In Richard Kronland-Martinet, Sølvi Ystad, and Kristoffer Jensen, editors, *Computer Music Modeling and Retrieval. Sense of Sounds*, number 4969 in Lecture Notes in Computer Science, pages 368–379. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85034-2 978-3-540-85035-9. URL http://link.springer.com/chapter/10.1007/978-3-540-85035-9_26.
- [68] Arne Eigenfeldt. The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine. In Mario Giacobini, Anthony Brabazon, Stefano Cagnoni, Gianni A. Di Caro, Anikó Ekárt, Anna Isabel Esparcia-Alcázar, Muddassar Farooq, Andreas Fink, and Penousal Machado, editors, *Applications of Evolutionary Computing*, number 5484 in Lecture Notes in Computer Science, pages 498–507. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01128-3 978-3-642-01129-0. URL http://link.springer.com/chapter/10.1007/978-3-642-01129-0_56.
- [69] Arne Eigenfeldt. Coming together: Composition by negotiation. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1433–1436. ACM, 2010. URL <http://dl.acm.org/citation.cfm?id=1874237>.
- [70] Arne Eigenfeldt. Multi-Agent Modeling of Complex Rhythmic Interactions in RealTime Performance. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.

- [71] Arne Eigenfeldt. Generating Structure—Towards Large-Scale Formal Generation. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. URL <http://musicalmetacreation.org/mume2014/content/proceedings/Generating%20Structure%20-%20Towards%20Large-scale%20Formal%20Generation.pdf>.
- [72] Arne Eigenfeldt. Exploring moment-form in generative music. In *Proceedings of 13th Sound and Music Conference*, Hamburg, Germany, 2016. ISBN 978-3-00-053700-4. URL https://www.researchgate.net/profile/Arne_Eigenfeldt/publication/306207035_EXPLORING_MOMENT-FORM_IN_GENERATIVE_MUSIC/links/57b35e5808aeac3177849721.pdf.
- [73] Arne Eigenfeldt. Musebots at One Year: A Review. In *Proceedings of the 4th International Workshop on Musical Metacreation (MUME 2016)*, 2016. ISBN 978-0-86491-397-5. URL https://www.researchgate.net/profile/Arne_Eigenfeldt/publication/306206920_Musebots_at_One_Year_A_Review/links/57b35df608aeaf239baf1456.pdf.
- [74] Arne Eigenfeldt and Philippe Pasquier. A realtime generative music system using autonomous melody, harmony, and rhythm agents. In *Proceedings of the 12th Generative Art Conference*, 2009. URL <http://artscience-ebookshop.com/on/cic/GA2009Papers/p7.pdf>.
- [75] Arne Eigenfeldt and Philippe Pasquier. Negotiated content: Generative soundscape composition by autonomous musical agents in Coming Together: Freesound. In *Proceedings of the Second International Conference on Computational Creativity, Mexico City*, pages 27–32, 2011. URL http://www.researchgate.net/profile/Arne_Eigenfeldt/publication/228411309_Negotiated_Content_Generative_Soundscape_Composition_by_Autonomous_Musical_Agents_in_Coming_Together_Freesound/links/0912f5093deae4b882000000.pdf.
- [76] Arne Eigenfeldt and Philippe Pasquier. A sonic eco-system of self-organising musical agents. In *9th European Event on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt 2011)*, volume 6625, pages 283–292, Torino, 2011. Springer Verlag.
- [77] Arne Eigenfeldt and Philippe Pasquier. Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. In *Proceedings of Sound and Music Computing 2012*, pages 181–186, Copenhagen, Denmark, 2012. URL <http://www.smcnetwork.org/system/files/smc2012-187.pdf>.
- [78] Arne Eigenfeldt, Oliver Bown, Philippe Pasquier, and Aengus Martin. Towards a Taxonomy of Musical Metacreation: Reflections on the First Musical Metacreation

- Weekend. In *Proceedings of the 2nd International Workshop on Musical Metacreation (MUME 2013)*, 2013. URL http://www.researchgate.net/profile/Arne_Eigenfeldt/publication/258258077_Towards_a_Taxonomy_of_Musical_Metacreation_Reflections_on_the_First_Musical_Metacreation_Weekend/links/02e7e527a2da4e5426000000.pdf.
- [79] Arne Eigenfeldt, Oliver Bown, and Benjamin Carey. Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble. In *Proceedings of the Sixth International Conference on Computational Creativity June*, page 134, 2015. URL <http://axon.cs.byu.edu/ICCC2015proceedings/6.2Eigenfeldt.pdf>.
- [80] Aaron Einbond, Riccardo Borghesi, Diemo Schwarz, and Norbert Schnell. Introducing CatOracle: Corpus-based concatenative improvisation with the Audio Oracle algorithm. In *Proceedings of the International Computer Music Conference 2016*, pages 140–146, 2016.
- [81] Mustafa Emirbayer and Ann Mische. What Is Agency? *American Journal of Sociology*, 103(4):962–1023, January 1998. ISSN 0002-9602, 1537-5390. doi: 10.1086/231294. URL <http://www.journals.uchicago.edu/doi/10.1086/231294>.
- [82] Jianyu Fan, Kivanç Tatar, Miles Thorogood, and Philippe Pasquier. Ranking-Based Emotion Recognition for Experimental Music. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) 2017*, 2017.
- [83] Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations: an organizational view of multi-agent systems. In *International Workshop on Agent-Oriented Software Engineering*, pages 214–230. Springer, 2003. URL http://link.springer.com/chapter/10.1007/978-3-540-24620-6_15.
- [84] Charles B. Fowler. The Museum of Music: A History of Mechanical Instruments. *Music Educators Journal*, 54(2):45, October 1967. ISSN 00274321. doi: 10.2307/3391092. URL <http://m ej.sagepub.com/cgi/doi/10.2307/3391092>.
- [85] Alexandre R. J. François, E. Chew, and Dennis Thurmond. Performer-centered visual feedback for human-machine improvisation. *Computers in Entertainment*, 9(3):1–13, November 2011. ISSN 15443574. doi: 10.1145/2027456.2027459. URL <http://dl.acm.org/citation.cfm?doid=2027456.2027459>.
- [86] Alexandre RJ François, Isaac Schankler, and Elaine Chew. Mimi4x: An interactive audio-visual installation for high-level structural improvisation. *International Journal of Arts and Technology*, 6(2):138–151, 2013. URL <http://www.inderscienceonline.com/doi/abs/10.1504/IJART.2013.053557>.

- [87] Christopher Frayling. Research in Art and Design (Royal College of Art Research Papers, Vol 1, No 1, 1993/4). *Royal College of Art Research Papers*, 1(1), 1994. URL <http://researchonline.rca.ac.uk/384/>.
- [88] Philip Galanter. What is generative art? Complexity theory as a context for art theory. In *Proceedings of the 6th Generative Art Conference*, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.2634>.
- [89] R.A. Garcia. *Automatic generation of sound synthesis techniques*. PhD thesis, MIT, 2001.
- [90] Dileep George. *How the brain might work: A hierarchical and temporal model for learning and recognition*. PhD thesis, Stanford University, 2008.
- [91] Toby Gifford. Appropriate and Complementary Rhythmic Improvisation in an Interactive Music System. In Simon Holland, Katie Wilkie, Paul Mulholland, and Allan Seago, editors, *Music and Human-Computer Interaction*, Springer Series on Cultural Computing. Springer London, London, 2013. ISBN 978-1-4471-2989-9 978-1-4471-2990-5. URL <http://link.springer.com/10.1007/978-1-4471-2990-5>.
- [92] Toby M. Gifford and Andrew R. Brown. Anticipatory timing in algorithmic rhythm generation. In *Proceedings of the Australasian Computer Music Conference 2010*, pages 21–28. Australasian Computer Music Association (ACMA), 2010. URL <http://eprints.qut.edu.au/33281/>.
- [93] Marco Gillies, Bongshin Lee, Nicolas d’Alessandro, Joëlle Tilmanne, Todd Kulesza, Baptiste Caramiaux, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, and Saleema Amershi. Human-Centred Machine Learning. pages 3558–3565. ACM Press, 2016. ISBN 978-1-4503-4082-3. doi: 10.1145/2851581.2856492. URL <http://dl.acm.org/citation.cfm?doid=2851581.2856492>.
- [94] Marcelo Gimenes and Eduardo Reck Miranda. An Ontomematic Approach to Musical Intelligence. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.
- [95] Marcelo Gimenes, Eduardo Reck Miranda, and Chris Johnson. Towards an intelligent rhythmic generator based on given examples: a memetic approach. In *Digital Music Research Network Summer Conference*, pages 41–46, 2005. URL http://cmr.soc.plymouth.ac.uk/publications/Gimenes_Glasgow_def.pdf.
- [96] Marcelo Gimenes, Eduardo Reck Miranda, and Chris Johnson. Musicianship for robots with style. In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, pages 197–202. ACM, 2007. URL <http://dl.acm.org/citation.cfm?id=1279778>.

- [97] Rich Gold and John Maeda. *The Plenitude : Creativity, Innovation, and Making Stuff*. The MIT Press, Cambridge, US, 2007. ISBN 978-0-262-27399-2. URL <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10205843>.
- [98] Antoni Gomila and Vincent C. Müller. Challenges for artificial cognitive systems. *Journal of Cognitive Science*, 13(4):453–469, 2012. URL <http://philpapers.org/rec/GOMCFA>.
- [99] Simon Harding, Jürgen Leitner, and Jürgen Schmidhuber. Cartesian Genetic Programming for Image Processing. In Rick Riolo, Ekaterina Vladislavleva, Marylyn D. Ritchie, and Jason H. Moore, editors, *Genetic Programming Theory and Practice X*, Genetic and Evolutionary Computation, pages 31–44. Springer New York, 2013. ISBN 978-1-4614-6845-5 978-1-4614-6846-2. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-1-4614-6846-2_3. DOI: 10.1007/978-1-4614-6846-2_3.
- [100] Arno Hartholt, David Traum, Stacy C. Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents*, Edinburgh, UK, August 2013. URL <http://ict.usc.edu/pubs/All%20Together%20Now.pdf>.
- [101] Andrew Hawryshkewich, Philippe Pasquier, and Arne Eigenfeldt. Beatback: A Real-time Interactive Percussion System for Rhythmic Practise and Exploration. *Proceedings of the tenth International Conference on New Interfaces for Musical Expression*, pages 100–105, 2010. URL <http://www.nime.org/>.
- [102] Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A Functional Taxonomy of Music Generation Systems. *ACM Computing Surveys*, 50(5):1–30, September 2017. ISSN 03600300. doi: 10.1145/3108242. URL <http://dl.acm.org/citation.cfm?doid=3145473.3108242>.
- [103] Francis Heylighen. Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research*, 38:4–13, June 2016. ISSN 13890417. doi: 10.1016/j.cogsys.2015.12.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S1389041715000327>.
- [104] William Hsu. Strategies for Managing Timbre and Interaction in Automatic Improvisation Systems. *Leonardo Music Journal*, 20(1):33–39, 2010. ISSN 1531-4812. URL <https://muse-jhu-edu.proxy.lib.sfu.ca/article/404089>.
- [105] David Brian Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, Cambridge, UNITED STATES, 2014. ISBN 978-0-262-27596-5. URL <http://ebookcentral.proquest.com/lib/sfu-ebooks/detail.action?docID=3338552>.

- [106] Anna Jordanous. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation*, 4(3):246–279, September 2012. ISSN 1866-9956, 1866-9964. doi: 10.1007/s12559-012-9156-1. URL <http://link.springer.com/10.1007/s12559-012-9156-1>.
- [107] John P. Kimball. *Syntax and Semantics*. Academic Press, 1975. ISBN 978-0-12-785423-6.
- [108] Alexis Kirke and Eduardo Miranda. A Biophysically Constrained Multi-Agent Systems Approach to Algorithmic Composition with Expressive Performance. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.
- [109] Alexis Kirke and Eduardo Miranda. A Multi-Agent Emotional Society Whose Melodies Represent its Emergent Social Hierarchy and Are Generated by Agent Communications. *Journal of Artificial Societies and Social Simulation*, 18(2):16, 2015. ISSN 1460-7425. doi: 10.18564/jasss.2679. URL <http://jasss.soc.surrey.ac.uk/18/2/16.html>.
- [110] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982. URL <http://link.springer.com/article/10.1007/BF00337288>.
- [111] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1–3):1–6, November 1998. ISSN 0925-2312. doi: 10.1016/S0925-2312(98)00030-7. URL <http://www.sciencedirect.com/science/article/pii/S0925231298000307>.
- [112] Olivier Lartillot, Donato Cereghetti, Kim Eliard, and Didier Grandjean. A simple, high-yield method for assessing structural novelty. In *Proceedings of the 3rd International Conference on Music & Emotion (ICME3), Jyväskylä, Finland, 11th-15th June 2013. Geoff Luck & Olivier Brabant (Eds.)*. ISBN 978-951-39-5250-1. University of Jyväskylä, Department of Music, 2013. URL <https://jyx.jyu.fi/dspace/handle/123456789/41611>.
- [113] Arnaud Lefebvre and Thierry Lecroq. A Heuristic For Computing Repeats With A Factor Oracle: Application To Biological Sequences. *International Journal of Computer Mathematics*, 79(12):1303–1315, January 2002. ISSN 0020-7160, 1029-0265. doi: 10.1080/00207160214653. URL <http://www.tandfonline.com/doi/abs/10.1080/00207160214653>.
- [114] Aaron Levisohn and Philippe Pasquier. BeatBender: subsumption architecture for autonomous rhythm generation. In *Proceedings of the ACM International Conference on Advances in Computer Entertainment Technologies (ACE 2008)*, pages 51–58, Yokohama, Japan, 2008. URL <http://eprints.iat.sfu.ca/883/>.

- [115] Benjamin Lévy, Georges Bloch, and Gérard Assayag. OMaxist dialectics. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 137–140, 2012. URL <https://hal.archives-ouvertes.fr/hal-00706662/>.
- [116] George E. Lewis. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10:33–39, 2000. URL <http://www.mitpressjournals.org/doi/abs/10.1162/096112100570585>.
- [117] Adam Linson, Chris Dobbyn, George E. Lewis, and Robin Laney. A Subsumption Agent for Collaborative Free Improvisation. *Computer Music Journal*, 39(4):96–115, 2015. ISSN 1531-5169. URL https://muse-jhu-edu.proxy.lib.sfu.ca/journals/computer_music_journal/v039/39.4.linson.html.
- [118] Michael F. Lynch. Motivation, Microdrives and Microgoals in Mockingbird. In *Proceedings of 3rd International Workshop on Musical Metacreation (MUME 2014)*, North Carolina, USA, 2014. URL <http://musicalmetacreation.org/mume2014/content/proceedings/Motivation,%20Microdrives%20and%20Microgoals%20in%20Mockingbird.pdf>.
- [119] M. Macret and P. Pasquier. Automatic Design of Sound Synthesizers As Pure Data Patches Using Coevolutionary Mixed-typed Cartesian Genetic Programming. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 309–316, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2662-9.
- [120] Aengus Martin and Oliver Bown. The Agent Designer Toolkit. In *Proceedings of the 9th ACM Conference on Creativity & Cognition*, C&C '13, pages 386–387, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2150-1. doi: 10.1145/2466627.2481211. URL <http://doi.acm.org/10.1145/2466627.2481211>.
- [121] Aengus Martin, C. T. Jin, André van Schaik, and William L. Martens. Partially observable Markov decision processes for interactive music systems. In *Proceedings of the International Computer Music Conference*, 2010. URL <http://www.ee.usyd.edu.au/carlab/CARlabPublicationsData/PDF/2010%20Martin%20In%20International%20Computer%20Music%20Conference-3237008059/2010%20Martin%20In%20International%20Computer%20Music%20Conference.pdf>.
- [122] Aengus Martin, Craig T. Jin, and Oliver Bown. A Toolkit for Designing Interactive Musical Agents. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, OzCHI '11, pages 194–197, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1090-1. doi: 10.1145/2071536.2071567. URL <http://doi.acm.org/10.1145/2071536.2071567>.

- [123] Aengus Martin, Craig T. Jin, and Oliver Bown. Implementation of a real-time musical decision-maker. In *Proceedings of the Australasian Computer Music Conference*, 2012. URL <http://www.maths.tcd.ie/~hobo/papers/2012acmc.pdf>.
- [124] Aengus Martin, Craig T. Jin, Ben Carey, and Oliver Bown. Creative experiments using a system for learning high-level performance structure in ableton live. In *Proceedings of the Sound and Music Computing Conference*, 2012. URL <http://smcnetwork.org/system/files/smc2012-206.pdf>.
- [125] Joao M. Martins and Eduardo R. Miranda. A connectionist architecture for the evolution of rhythms. In *Applications of Evolutionary Computing*, pages 696–706. Springer, 2006. URL http://link.springer.com/chapter/10.1007/11732242_66.
- [126] Joao M. Martins and Eduardo R. Miranda. Emergent rhythmic phrases in an A-Life environment. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, pages 10–14, 2007. URL http://cmr.soc.plymouth.ac.uk/publications/MusicAL_Martins.pdf.
- [127] Joao M. Martins and Eduardo R. Miranda. Breeding rhythms with artificial life. In *Proceedings of the Sound and Music Conference*. Citeseer, 2008. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.3253&rep=rep1&type=pdf>.
- [128] James B. Maxwell, Philippe Pasquier, and Eigenfeldt Eigenfeldt. Hierarchical Sequential Memory for Music: A Cognitive Model. In *Proceedings of the 10th International Conference for Music Information Retrieval*, 2009.
- [129] James B. Maxwell, Arne Eigenfeldt, Philippe Pasquier, and N. Gonzalez Thomas. MusiCOG: A cognitive architecture for music learning and generation. In *Proceedings of the Sound and Music Computing Conference*, page 9, 2012. URL <http://smcnetwork.org/system/files/smc2012-255.pdf>.
- [130] Jon McCormack and Oliver Bown. Life's what you make: Niche construction and evolutionary art. In *Workshops on Applications of Evolutionary Computation*, pages 528–537. Springer, 2009. URL http://link.springer.com/chapter/10.1007/978-3-642-01129-0_59.
- [131] Jon McCormack, Peter McIlwain, Aidan Lane, and Alan Dorin. Generative composition with Nodal. In *Workshop on music and artificial life (part of ECAL 2007), Lisbon, Portugal*, 2007. URL http://www.researchgate.net/profile/Aidan_Lane/publication/228749312_Generative_composition_with_Nodal/links/004635289f6dc0d7d6000000.pdf.

- [132] Julian F. Miller, editor. *Cartesian Genetic Programming*. Natural Computing Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-17309-7 978-3-642-17310-3. URL <http://link.springer.com/10.1007/978-3-642-17310-3>.
- [133] Marvin Minsky. *The society of mind*. Simon and Schuster, New York, N.Y, 1986. ISBN 978-0-671-60740-1.
- [134] Eduardo R. Miranda, Alexis Kirke, and Qijun Zhang. Artificial Evolution of Expressive Performance of Music: An Imitative Multi-Agent Systems Approach. *Computer Music Journal*, 34(1):80–96, 2010. ISSN 0148-9267. URL <http://www.jstor.org.proxy.lib.sfu.ca/stable/25653532>.
- [135] Eduardo Reck Miranda and Al Biles, editors. *Evolutionary computer music*. Springer, London, 2007. ISBN 978-1-84628-599-8. OCLC: ocm80332658.
- [136] Tom Michael Mitchell. *Machine Learning*. McGraw-Hill Education, March 1997. ISBN 978-0-07-042807-2.
- [137] Julian Moreira, Pierre Roy, and François Pachet. Virtualband: Interacting with Stylistically Consistent Agents. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 341–346, Brazil, 2013. URL http://ismir2013.ismir.net/wp-content/uploads/2013/09/277_Paper.pdf.
- [138] Meinard Müller. *Fundamentals of Music Processing*. Springer International Publishing, Cham, 2015. ISBN 978-3-319-21944-8 978-3-319-21945-5. URL <http://link.springer.com/10.1007/978-3-319-21945-5>.
- [139] D. Murray-Rust, A. Smaill, and M.C. Maya. VirtuaLatin - towards a musical multi-agent system. In *Sixth International Conference on Computational Intelligence and Multimedia Applications, 2005*, pages 17–22, August 2005. doi: 10.1109/ICCIMA.2005.59.
- [140] Dave Murray-Rust and Alan Smaill. Towards a model of musical interaction and communication. *Artificial Intelligence*, 175(9-10):1697–1721, June 2011. ISSN 00043702. doi: 10.1016/j.artint.2011.01.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S0004370211000038>.
- [141] David Murray-Rust. *Musical Acts and Musical Agents: theory, implementation and practice*. PhD thesis, 2008. URL <https://www.era.lib.ed.ac.uk/handle/1842/2561>.
- [142] David Murray-Rust, Alan Smaill, and Michael Edwards. MAMA: An Architecture for Interactive Musical Agents. *Frontiers in Artificial Intelligence and Applications*, 141:36, 2006.
- [143] DS Murray-Rust and Alan Smaill. Musical acts and musical agents. *Proceedings of the 5th Open Workshop of MUSICNETWORK: Integration of Music in Multimedia Applications (to Appear)*, 10, 2005.

- [144] Maria Navarro, Juan Manuel Corchado, and Yves Demazeau. A Musical Composition Application Based on a Multiagent System to Assist Novel Composers. *International Conference on Computational Creativity*, 2014.
- [145] Maria Navarro, Juan Manuel Corchado, and Yves Demazeau. MUSIC-MAS: Modeling a harmonic composition system with virtual organizations to assist novice composers. *Expert Systems with Applications*, 57:345–355, September 2016. ISSN 09574174. doi: 10.1016/j.eswa.2016.01.058. URL <http://linkinghub.elsevier.com/retrieve/pii/S0957417416300227>.
- [146] Jérôme Nika and Marc Chemillier. Improtak: integrating harmonic controls into improvisation in the filiation of OMax. In *International Computer Music Conference (ICMC)*, pages 180–187, 2012. URL <https://hal.archives-ouvertes.fr/hal-01059330/>.
- [147] Jérôme Nika, José Echeveste, Marc Chemillier, and Jean-Louis Giavitto. Planning Human-Computer Improvisation. In *International Computer Music Conference*, page 330, 2014. URL <https://hal.archives-ouvertes.fr/hal-01053834/>.
- [148] Jérôme Nika, Dimitri Bouche, Jean Bresson, Marc Chemillier, and Gérard Assayag. Guided improvisation as dynamic calls to an offline model. In *Sound and Music Computing (SMC)*, Maynooth, Ireland, July 2015. URL <https://hal.archives-ouvertes.fr/hal-01184642>.
- [149] Jérôme Nika, Marc Chemillier, and Gérard Assayag. ImprotK: Introducing Scenarios into Human-Computer Music Improvisation. *Computers in Entertainment*, 14(2):1–27, January 2017. ISSN 15443574. doi: 10.1145/3022635. URL <http://dl.acm.org/citation.cfm?doid=3023311.3022635>.
- [150] Doug Van Nort, Pauline Oliveros, and Jonas Braasch. Electro/Acoustic Improvisation and Deeply Listening Machines. *Journal of New Music Research*, 42(4):303–324, December 2013. ISSN 0929-8215. doi: 10.1080/09298215.2013.860465. URL <http://dx.doi.org/10.1080/09298215.2013.860465>.
- [151] François Pachet. Rhythms as emerging structures. In *Proceedings of 2000 International Computer Music Conference, Berlin, ICMA*, 2000. URL http://www.researchgate.net/profile/Francois_Pachet/publication/243602024_Rhythms_as_emerging_structures/links/0deec52909fbb73f05000000.pdf.
- [152] François Pachet. Beyond the cybernetic jam fantasy: The continuator. *Computer Graphics and Applications, IEEE*, 24(1):31–35, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1255806.

- [153] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003. URL <http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.3.333.16861>.
- [154] Ana Paiva, Gerd Andersson, Kristina Höök, Dário Mourão, Marco Costa, and Carlos Martinho. Sentoy in fantasy: Designing an affective sympathetic interface to a computer game. *Personal and Ubiquitous Computing*, 6(5-6):378–389, 2002. URL <http://dl.acm.org/citation.cfm?id=592611>.
- [155] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An Introduction to Musical Metacreation. *Computers in Entertainment*, 14(2):1–14, January 2017. ISSN 15443574. doi: 10.1145/2930672. URL <http://dl.acm.org/citation.cfm?doid=3023311.2930672>.
- [156] Alison Pease and Simon Colton. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*, 2011.
- [157] M. Peter. Milieus of creativity: The role of places, environments, and spatial. In Peter Meusburger, Joachim Funke, and Edgar Wunder, editors, *Milieus of creativity: an interdisciplinary approach to spatiality of creativity*, number v. 2 in *Knowledge and space*, pages 97–153. Springer, Dordrecht : [Heidelberg], 2009. ISBN 978-1-4020-9876-5.
- [158] David Plans and Davide Morelli. Using Coevolution in Music Improvisation. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.
- [159] Reinier Plomp and Willem Johannes Maria Levelt. Tonal consonance and critical bandwidth. *The journal of the Acoustical Society of America*, 38(4):548–560, 1965. URL <http://asa.scitation.org/doi/abs/10.1121/1.1909741>.
- [160] Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and mathematical statistics. Applied probability and statistics section. John Wiley & Sons, New York, 1994. ISBN 978-0-471-61977-2.
- [161] Jaume Rigau, Miquel Feixas, and Mateu Sbert. Informational aesthetics measures. *IEEE Computer Graphics and Applications*, 28(2):24–34, 2008. URL https://www.researchgate.net/profile/Mateu_Sbert/publication/5501365_Informational_Aesthetics_Measures/links/0912f51086574986ad000000.pdf.
- [162] Graeme Ritchie. Evaluating Quality in Creative Systems, 2014. URL http://videolectures.net/ascc2013_ritchie_systems/.

- [163] Curtis Roads. *Composing electronic music: a new aesthetic*. Oxford University Press, Oxford, 2015. ISBN 978-0-19-537324-0.
- [164] Robert Rowe. Machine Listening and Composing with Cypher. *Computer Music Journal*, 16(1):43, 1992. ISSN 01489267. doi: 10.2307/3680494. URL <http://www.jstor.org/stable/3680494?origin=crossref>.
- [165] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, December 1980. ISSN 0022-3514. doi: 10.1037/h0077714. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1981-25062-001&site=ehost-live>.
- [166] Stuart J. (Stuart Jonathan) Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, Upper Saddle River, N.J, 3rd edition, 2010. ISBN 978-0-13-207148-2.
- [167] Pablo Azevedo Sampaio, Geber Ramalho, and Patrícia Tedesco. CinBalada: a multiagent rhythm factory. *Journal of the Brazilian Computer Society*, 14(3):31–49, 2008. URL http://www.scielo.br/scielo.php?pid=S0104-65002008000300004&script=sci_arttext&tlang=es.
- [168] Herbert A. Simon. *The new science of management decision*, volume xii of *The Ford distinguished lectures*. Harper & Brothers, New York, NY, US, 1960. DOI: 10.1037/13978-000.
- [169] S. N. Sivanandam and S. N. Deepa. *Introduction to genetic algorithms*. Springer, Berlin ; New York, 2007. ISBN 978-3-540-73189-4.
- [170] Denis Smalley. Spectromorphology: explaining sound-shapes. *Organised Sound*, 2(02):107–126, August 1997. ISSN 1469-8153. doi: 10.1017/S1355771897009059. URL http://journals.cambridge.org/article_S1355771897009059.
- [171] Benjamin D. Smith and W. Scott Deal. ML.* Machine Learning Library as a Musical Partner in the Computer-Acoustic Composition Flight. In *the Proceedings of the Joint Conference ICMC14-SMC14*, volume 2014, Athens, Greece, 2014. URL <http://www.smc-conference.net/smc-icmc-2014/images/proceedings/OS19-B09-ML.pdf>.
- [172] Benjamin D. Smith and Guy E. Garnett. Reinforcement Learning and the Creative, Automated Music Improviser. In Penousal Machado, Juan Romero, and Adrian Carballal, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, number 7247 in Lecture Notes in Computer Science, pages 223–234. Springer Berlin Heidelberg, April 2012. ISBN 978-3-642-29141-8 978-3-642-29142-5. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-642-29142-5_20.

- [173] David J.T Sumpter and Madeleine Beekman. From nonlinearity to optimality: pheromone trail foraging by ants. *Animal Behaviour*, 66(2):273–280, August 2003. ISSN 00033472. doi: 10.1006/anbe.2003.2224. URL <http://linkinghub.elsevier.com/retrieve/pii/S000334720392224X>.
- [174] Greg Surges and Shlomo Dubnov. Feature selection and composition using PyOracle. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013. URL http://www.pucktronix.com/media/papers/Surges_Dubnov_MuME2013_final.pdf.
- [175] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. Adaptive Computation and Machine Learning. A Bradford Book, Cambridge, Mass, 1998. ISBN 978-0-262-19398-6. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1094&site=ehost-live>.
- [176] T. Takala, J. Hahn, L. Gritz, J. Geigel, and J. Lee. Using physically based models and genetic algorithms for functional composition of sound signals, synchronized to animated motion. In *Proceedings of the International Computer Music Conference*, pages 180–185, 1993.
- [177] Kivanç Tatar and Philippe Pasquier. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, June 2017. ISBN 978-1-77287-019-0.
- [178] Kivanç Tatar, Matthieu Macret, and Philippe Pasquier. Automatic Synthesizer Preset Generation with PresetGen. *Journal of New Music Research*, 45(2):124–144, April 2016. ISSN 0929-8215. doi: 10.1080/09298215.2016.1175481. URL <http://dx.doi.org/10.1080/09298215.2016.1175481>.
- [179] Kivanç Tatar, Philippe Pasquier, and Remy Siu. REVIVE: An Audio-visual Performance with Musical and Visual AI Agents. pages 1–6. ACM Press, 2018. ISBN 978-1-4503-5621-3. doi: 10.1145/3170427.3177771. URL <http://dl.acm.org/citation.cfm?doid=3170427.3177771>.
- [180] Belinda Thom. BoB: An Interactive Improvisational Music Companion. In *Proceedings of the Fourth International Conference on Autonomous Agents*, AGENTS '00, pages 309–316, New York, NY, USA, 2000. ACM. ISBN 1-58113-230-1. doi: 10.1145/336595.337510. URL <http://doi.acm.org/10.1145/336595.337510>.
- [181] Belinda Thom. Unsupervised learning and interactive jazz/blues improvisation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 652–657, 2000. URL <http://www.aaai.org/Papers/AAAI/2000/AAAI00-100.pdf>.

- [182] Belinda Thom. Interactive improvisational music companionship: A user-modeling approach. *User Modeling and User-Adapted Interaction*, 13(1-2):133–177, 2003. URL <http://link.springer.com/article/10.1023/A:1024014923940>.
- [183] Nicolas Gonzalez Thomas, Philippe Pasquier, Arne Eigenfeldt, and James B. Maxwell. A Methodology for the Comparison of Melodic Generation Models Using Meta-Melo. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 561–566, Brazil, 2013. ISBN 978-0-615-90065-0. URL http://ismir2013.ismir.net/wp-content/uploads/2013/09/228_Paper.pdf.
- [184] Kristinn Thórisson and Helgi Helgasson. Cognitive Architectures and Autonomy: A Comparative Review. *Journal of Artificial General Intelligence*, 3(2):1–30, January 2012. ISSN 1946-0163. doi: 10.2478/v10229-011-0015-3. URL <http://www.degruyter.com/view/j/jagi.2012.3.issue-2/v10229-011-0015-3.xml>.
- [185] Peter M Todd and Gregory M Werner. Frankensteinian methods for evolutionary music. In *Musical networks: parallel distributed perception and performance*, pages 313–340. MIT Press/Bradford Books, Cambridge, MA, 1999.
- [186] Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.
- [187] Leo Kazuhiro Ueda and Fabio Kon. Andante: A mobile musical agents infrastructure. In *Proceedings of the 9th Brazilian Symposium on Computer Music*, pages 87–94, 2003. URL http://gsd.ime.usp.br/sbcm/2003/papers/rLeo_Ueda.pdf.
- [188] Rafael Valle, Alexandre Donzé, Daniel J. Fremont, Ilge Akkaya, Sanjit A. Seshia, Adrian Freed, and David Wessel. Specification Mining for Machine Improvisation with Formal Specifications. *Computers in Entertainment*, 14(3):1–20, January 2017. ISSN 15443574. doi: 10.1145/2967504. URL <http://dl.acm.org/citation.cfm?doid=3023312.2967504>.
- [189] Edgard Varese and Chou Wen-chung. The liberation of Sound. *Perspectives of New Music*, 5(1):11–19, 1966. URL https://www.jstor.org/stable/832385?origin=JSTOR-pdf&seq=1#page_scan_tab_contents.
- [190] Rosa Maria Vicari, Lauro Nakayama, Rodolfo Daniel Wulffhorst, Leandro Lesqueves Costa-longa, and Evandro Manara Miletto. The Musical Interactions within Community Agents. *Agent-Based Simulation Conference*, 2005.
- [191] J. N. Vold. *A study of musical problem solving behavior in kindergarten children and a comparison with other aspects of creative behavior*. PhD thesis, University of Alabama, 1986.

- [192] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. URL http://acsweb.ucsd.edu/~chw160/pdf/mume2014_vmo_Wang_Dubnov.pdf.
- [193] Cheng-i Wang and Shlomo Dubnov. Context-Aware Hidden Markov Models of Jazz Music with Variable Markov Oracle. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, 2017.
- [194] Peter R. Webster. Conceptual Bases for Creative Thinking in Music. In *Music and Child Development*, pages 158–174. Springer, New York, NY, 1987. ISBN 978-1-4613-8700-8 978-1-4613-8698-8. URL https://link-springer-com.proxy.lib.sfu.ca/chapter/10.1007/978-1-4613-8698-8_8. DOI: 10.1007/978-1-4613-8698-8_8.
- [195] Kaare Wehn. Using Ideas from Natural Selection to Evolve Synthesized Sounds. In *Proceedings of the Digital Audio Effects DAFX98 workshop*, pages 159–167, Barcelona, 1998.
- [196] Gerhard Weiss. *Multiagent systems*. Intelligent robotics and autonomous agents. The MIT Press, Cambridge, Massachusetts, second edition. edition, 2013. ISBN 978-0-262-01889-0.
- [197] Ian Whalley. PIWeCS: enhancing human/machine agency in an interactive composition system. *Organised Sound*, 9(02), August 2004. ISSN 1355-7718, 1469-8153. doi: 10.1017/S135577180400024X. URL http://www.journals.cambridge.org/abstract_S135577180400024X.
- [198] Mitchell Whitelaw. *Metacreation: art and artificial life*. MIT Press, Cambridge, Mass, 2004. ISBN 9780262232340.
- [199] Geraint A. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7):449–458, November 2006. ISSN 09507051. doi: 10.1016/j.knosys.2006.04.009. URL <http://linkinghub.elsevier.com/retrieve/pii/S0950705106000645>.
- [200] Geraint A. Wiggins. Searching for computational creativity. *New Generation Computing*, 24(3):209–222, September 2006. ISSN 0288-3635, 1882-7055. doi: 10.1007/BF03037332. URL <http://link.springer.com/10.1007/BF03037332>.
- [201] Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, June 2009. ISBN 9780470519462.
- [202] Rodolfo Daniel Wulffhorst, Lauro Nakayama, and Rosa Maria Vicari. A Multiagent Approach for Musical Interactive Systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 584–591, New York,

- NY, USA, 2003. ACM. ISBN 1-58113-683-8. doi: 10.1145/860575.860669. URL <http://doi.acm.org/10.1145/860575.860669>.
- [203] Matthew Yee-King and Mark d’Inverno. Experience driven design of creative systems. In *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016). Universite Pierre et Marie Curie*, 2016. URL <http://www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/Experience-Driven-Design-of-Creative-Systems.pdf>.
- [204] Matthew John Yee-King. An Automated Music Improviser Using a Genetic Algorithm Driven Synthesis Engine. In Mario Giacobini, editor, *Applications of Evolutionary Computing*, number 4448 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-71804-8 978-3-540-71805-5. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-540-71805-5_62.
- [205] Michael Young. NN music: improvising with a ‘living’ computer. In *International Symposium on Computer Music Modeling and Retrieval*, pages 337–350. Springer, 2007. URL http://link.springer.com/chapter/10.1007/978-3-540-85035-9_23.

Chapter 3

MASOM: A Musical Agent Architecture based on Self-Organizing Maps, Affective Computing, and Variable Markov Models

KIVANÇ TATAR
PHILIPPE PASQUIER

AS PUBLISHED IN PROCEEDINGS OF THE 5TH INTERNATIONAL WORKSHOP ON MUSICAL METACREATION (MUMe 2017)

Abstract

Musical Agent based on Self-Organizing Maps (MASOM) is a machine improvisation software for live performance. MASOM plays experimental music and free improvisation. The agent perceives and generates audio signals. MASOM combines Self-Organizing Maps for sound memory, Variable Markov Models for musical structure, and Affective Computing for machine listening. The agent learns the sonic content and the musical structure to generate live performances. MASOM's offline learning uses an audio corpus of recordings of performances or compositions. The machine listening module of MASOM extracts high-level features such as eventfulness, pleasantness, and timbre. The agent listens to itself and other performers to decide what to play next.

Keywords: Musical agents; Multi-Agent Systems; Artificial Intelligence; Musical Metacreation; Computational Creativity; Virtual Reality; Interactive Arts; Contemporary Arts

3.1 Introduction

Metacreation is the idea of endowing machines with creative behaviors [25]. Metacreation applies the knowledge of Artificial Intelligence to develop autonomous systems solving creative tasks. *Musical Metacreation* (MUME) is a sub-branch of Metacreation. MUME focuses on the creative tasks of music. Autonomy and agency being essential components of Metacreation and MUME, artificial agents become the perfect modeling paradigm.

In this study, we present a new musical agent architecture. An agent is a proactive system that autonomously initiates actions to respond to its environment in timely fashion [36]. Agents work both online and offline. While a variety of musical tasks have been addressed by Multi-agent Systems (MAS), we focus on improvisation in experimental electronic music in this study.

A *musical agent* is an autonomous system that creates music or a part of the music, individually or in a community of agents. MASOM is a flexible musical agent that only requires a corpus of recordings for the learning and an audio signal as an input to listen to other agents. MASOM implements a hybrid agent architecture that combines Self-Organizing Maps as a musical memory, Variable Order Markov Models for pattern recognition and generation in music, and Affective Computing and machine listening to model human hearing.

MASOM's architecture stands out with the following contributions in musical agents:

- The use of SOMs as a musical memory of audio samples
- The capacity of a musical agent that can listen to big data of music
- The introduction of sound affect estimation in offline and online machine listening
- The flexibility of a musical agent to perform alone or with other agents, software or human.
- Machine listening with the time scales of *micro*, *sound object*, and *meso*

Roads [28] proposes *infinitesimal*, *subsample*, *sample*, *micro*, *sound object*, *meso*, *macro*, *supra*, and *infinite* time scales of music, arranged from the one with the shortest duration to the longest one. MASOM inherits three different time scales of *micro*, *sound object*, and *meso*. *Micro* time scale spans from a millisecond to approximately 100ms. The duration of sound objects varies from a fraction of a second to several seconds. *Meso* time scale ranges from seconds to minutes. We explain how these time scales are incorporated in MASOM's architecture in the Section ??.

3.2 Background

3.2.1 Self-Organizing Maps

Self-Organizing Maps (SOMs) are artificial neural network models, inspired by neurophysiology [15]. SOMs visualize, represent, and cluster high-dimensional input data with a simpler 2D topology. SOM topologies are typically square and include a finite number of nodes. Node vectors have the

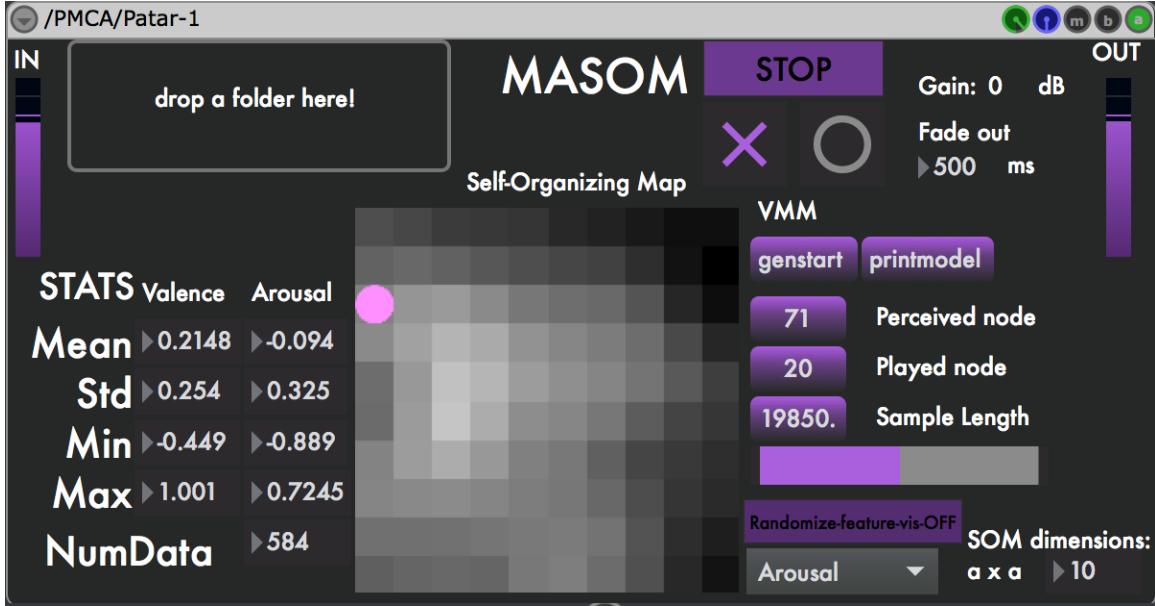


Figure 3.1: The generation interface of MASOM includes a visualization of SOM. The visualization shows one dimension at a time. The dimensions are normalized between -1.0 (black) and 1.0 (white) for the visualization.

same number of dimensions as the input data. SOMs organize the input data using a 2D similarity grid so that similar data clusters locate closer to each other in the topology. Moreover, SOMs cluster the input data by assigning each input vector to the closest node called the best matching unit (BMU). Figure 3.1 shows a SOM with 100 nodes. Each square represents one node that is colored according to its feature value.

The training is unsupervised in SOMs, but designers set the topology and the number of nodes in the topology. Each input vector is a training instance of SOM's learning. There are three ways to initialize SOM nodes: starting with zero vectors, randomizing model vectors, and using principal component analysis on the input vectors. During a training instance, a SOM also updates BMU's neighboring model vectors using a *neighborhood function*. Common *neighborhood functions* are Gaussian, cut-Gaussian curves, linear, and piecewise linear functions.

On each training instance, SOMs update their nodes using the data of an input vector. First, SOMs find the BMU of an input vector. Second, SOMs calculate the Euclidean distance between the input vector and the BMU. Third, SOMs update the BMU by this distance multiplied by the *learning rate*. The *learning rate* is a user-set global parameter in the range [0., 1.]. Lower learning rate corresponds to less adaptive and more history-depended SOMs. Depending on the neighboring function, SOM also updates the neighbors of BMU in the direction of BMU's update. The update amount becomes less as the neighboring node is further away from the BMU. Therefore, the BMU and its neighboring nodes move closer to input vectors on each training instance.

It is common to use the same dataset more than once to train a SOM. Each pass on the dataset is called an *epoch*. The learning rate is adaptive, decreasing as the number of epoch increases. Hence,

model vectors change less as the number of epochs increases [15]. The training of SOM stops after some epochs, or when the nodes are updated less than a user-set amount, or not updated at all.

3.2.2 Markov Models

Markov Models are finite state machines that model patterns in discrete sequences through the Markov assumption. An N^{th} order Markov model assumes that,

$$P(s_t|s_{t-1}, s_{t-2}, \dots, s_1) = P(s_t|s_{t-1}, \dots, s_{\max(t-N, 1)}) \quad (3.1)$$

Hence, Markov models are history dependent. A first order Markov model depends on the current state to predict the next. A second order Markov model takes the current and the previous state into account to predict the next state. That is, the order of a Markov model points out how many previous states to be considered to predict and generate the next state. Moreover, the observed number of transitions between Markov states determines the conditional probabilities of the transitions. Therefore, a Markov Model is a stochastic model represented as a directional graph.

Variable Markov models (VMMs) refer to a family of algorithms including probabilistic finite automata, probabilistic suffix automata, prediction suffix trees, Lempel-Ziv 78, improved Lempel-Ziv (LZ-MS), prediction by partial match (PPM), Factor Oracles, and the context tree weighting method [29, 3]. VMM considers a varying number of previous states to predict the next state. Markov Models are applied to a variety of MUME tasks including the musical agent design (see Section Musical Agents with Markov Models).

3.2.3 Affective Computing

Erola and Vuoskoski [8] mention that “...the emotional effects of music are the most important reason why people engage in musical activities.” and categorize affect models in the literature in four classes: *discrete*, *dimensional*, *miscellaneous* and *music-specific*. First, the discrete affect models assume that one can explain all emotions using a finite set of basic emotions such as happiness, sadness, fear, anger, and disgust [10, 24] as well as shame, embarrassment, contempt and guilt [22]. Second, the dimensional affect models represent emotions using two or more continuous dimensions that are ideally orthogonal to each other. The most common dimensional affect model has two dimensions: valence (pleasantness) and arousal (eventfulness). Some dimensional affect models include additional dimensions such as tension, potency, dominance [7]. The continuous circumplex model is an example of a dimensional affect model [30]. Music Information Retrieval (MIR) studies frequently use the dimensional affect model [8]. Third, miscellaneous models are collections of concepts that are linked to emotions such as intensity, preference, similarity, tension. Fourth, music-specific affect models are proposed as emotion lists that are specifically relevant to music [37]. The discussions around a list of music-specific emotions are still ongoing [8].

The cognition of affect has different layers. Livingstone et al. [18] model three different layers in the cognition of emotion in sound and music: perceived, induced and expressed emotion. The perceived emotion is the subjective perception of stimuli. The perceived emotion goes through a

cognitive process and becomes the reaction that is the induced emotion. The expressed emotion is the conveyed emotion that is stimulated by subjects (humans). In this study, the affect estimation in sound focuses on the perceived emotion. We explain our multivariate regression algorithm for affective computing in sound in the Section System Design.

3.3 Related Work

In this section, we explain why we choose SOMs to model musical memory and mention a musical agent that use a SOM in the system design. Then, we briefly cover musical agents with Variable Markov Models.

3.3.1 Modelling Musical Memory with SOMs

Gabora [13] proposes three properties of the memory in the cognitive processes of creativity:

- Memory is sparse
- Memory is distributed, but distributions are constrained
- Memory is content addressable

Bogart and Pasquier [4] build on Gabora's work to model the memory of creative visual processes using SOMs. Bogart and Pasquier propose that SOMs are beneficial to model the connection between the sensory input and the field experience of an agent. SOMs satisfy the three memory properties of creative processes proposed by Gabora. If we model the creative memory with SOM, the memory is sparse since SOMs consist of separate node vectors arranged in a 2D plane. The memory is distributed but distributions are constrained because SOMs have a finite number of nodes that are constrained by the domain represented by the sensory input. Memory is also content addressable in SOMs. SOM nodes represent the clusters the sensory input instances. Moreover, the node vectors are not the exact replications of sensory input vectors although SOM nodes are aligned by the sensory input.

Regarding the applications of SOMs as the memory of a musical agent, we have found only one implementation in the literature. Smith and Deal [31] use SOMs in the short term memory of a musical agent. This musical agent works with audio inputs and extracts audio features of chroma, brightness, noisiness, and loudness. There are two layers in the memory: long-term memory and adaptive memory. The long-term memory is a k-d tree trained on audio feature vectors at the end of each performance sessions. Also, each input audio feature vector is a search query of the k-d tree during the performance. The agent trains and updates the SOM online using chroma vectors of the input. The amount of change in SOM node vectors in a control signal that is passed to the *decision* module. The decision module uses this distance to deviate from the input feature vector to introduce variance in the agent's output.

SOMs have also been used to organize large collections of audio samples [9, 12]. Hence, we decided to focus on SOMs as the sound object memory of MASOM.

3.3.2 Musical Agents with Markov Models

Markov Models have been extensively used in musical agent design because of their success on the prediction [3] and generation [23] of symbolic music sequences.

The *Continuator* is a musical agent working with a symbolic representation of music [23]. The Continuator uses VMM to continue a musical phrase. Another musical agent with VMM is *Beatback* [14]. Beatback uses VMM to generate rhythms. Moreover, Factor Oracle, an algorithm that is similar to VMM, has been extensively applied to musical agents design, including *OMAX*, *Audio Oracle*, *PyOracle*, *Improtak*, and *Variable Markov Oracles* (VMO) systems [1, 2, 6, 17, 20, 33, 34, 21, 35]. Amongst all these musical agents, VMO is the closest architecture to MASOM. Similar to MASOM, VMO can perform live with or without other human or software agents. VMO differs from previous FO implementations. In the previous FO implementations, each state represents a musical phrase segment that is symbolic. However, each state in VMO is a cluster of audio frames. VMO uses a distance *threshold* to cluster audio frames. If the distance between feature vectors of two audio frames is less than the threshold, these frames are added to the same cluster. The agent sets the distance threshold automatically by calculating the threshold value that gives the highest *information rate* (IR). IR is extensively used in Pattern Matching and Recognition studies as a measure of information content.

3.4 System Design

This section begins with the explanation of the affect estimation algorithm and machine listening in MASOM. We continue by presenting the learning and generation in MASOM.

3.4.1 Sound Affect Estimation

Affect estimation in sound and music is still an open problem [8]. Fan et al. [11] present a machine learning model that estimates pleasantness and eventfulness of soundspace recordings. The authors implement a 2D continuous affect model proposed by Russell [30]. Using multivariate linear regression, their model is trained on a data set of 125 soundscape samples with 6-second duration. The data set is labeled by an online study with 20 participants. In this study, we use the same dataset that the study of Fan et al. [11] uses. We used a different audio feature extraction library and we applied multivariate regression to generate an affect estimation model. We implement feature extraction in MAX¹ using *ircamdescriptor*~ object provided in MAX Sound Box externals². MAX provides opportunities to use the affect estimation system for both offline affect estimation of an audio corpus and realtime affect estimation in machine listening applications. Next, we explain each audio feature used in the affect estimation model and introduce the model.

¹<https://cycling74.com/>

²<http://forumnet.ircam.fr/shop/en/forumnet/53-max-sound-box.html>.

All audio features are computed with a window size of 1024 samples (23ms) and a hop size of 512 samples (12ms), which correspond to the *micro* time scale. We calculate the mean and standard deviation of these audio features over a moving window of 6-seconds [11] which corresponds to the sound object time scale. There are five audio features included in the affect estimation model:

- *Mel Frequency Cepstral Coefficient* (MFCC) is a known feature in audio processing [26]. MFFC calculation combines Mel frequency scale with a particular frequency spectrum calculation called *cepstrum*. Mel frequency scale represents the critical bands of human hearing. The cepstrum stands for the discrete cosine transform (DCT) of the logarithm of the spectrum (FFT). There are 13 MFCCs in our calculation excluding the zero coefficient. *MFCC0*, the energy, or the DC offset is removed.
- The second feature that we use in the affect estimation is *loudness*. We use the algorithm proposed by Moore et al. [19] to calculate the loudness.
- *Spectral Flatness* is the ratio of geometric mean to the arithmetic mean of the energy spectrum. Spectral flatness shows the noisiness against sinusoidality of the spectrum. We compute the spectral flatness in four bands: 250 - 500, 500 - 1000, 1000 - 2000, 2000 - 4000 Hz. *SpectralFlatness1Mean* in equation 3.2a refers to the moving average of the spectral flatness computer over the band 250 - 500 Hz.
- *Perceptual Spectral Decrease* is the amount of decreasing of the spectral amplitude, computed using a human hearing model [26].
- *Tristimulus* is the calculation of three different types of energy ratio [26]. *Perceptual Tristimulus* uses a human hearing model to calculate tristimulus.

Equation 3.2a and 3.2b introduce our affect estimation model generated by the multivariate linear regression:

$$\begin{aligned}
 \text{Valence} = & -0.169 + & (3.2a) \\
 & - 0.061 * \text{LoudnessMean} \\
 & + 0.588 * \text{SpectralFlatness1Mean} \\
 & + 0.302 * \text{MFCC1STD} \\
 & + 0.361 * \text{MFCC5STD} \\
 & - 0.229 * \text{PerceptualSpectralDecreaseSTD}
 \end{aligned}$$

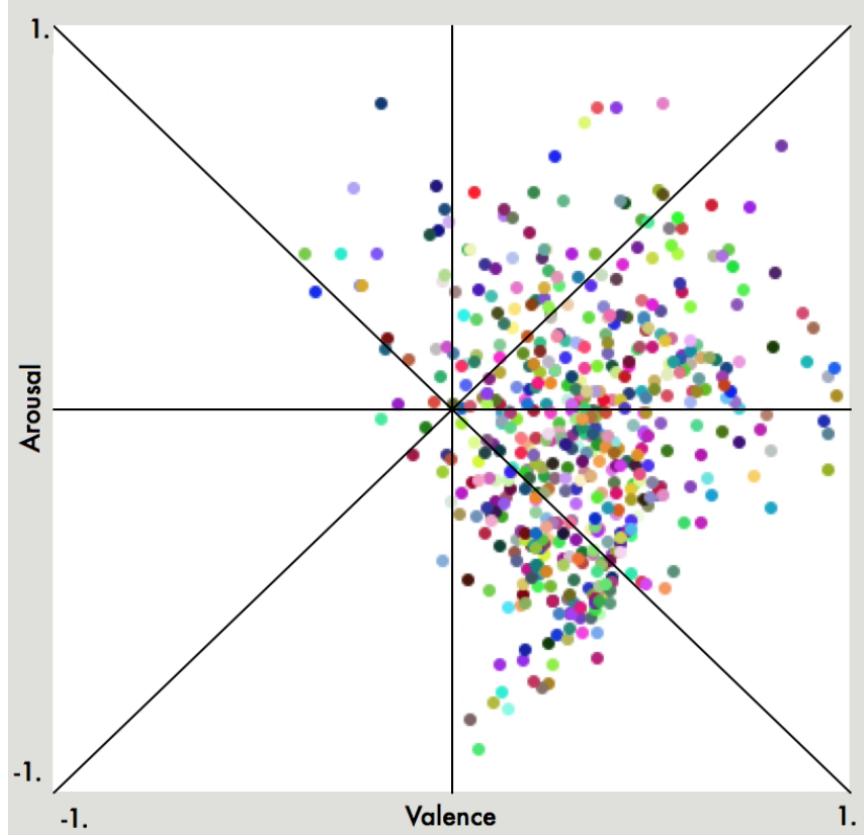


Figure 3.2: Each dot represents a novel segment of Stockhausen's *Kontakte*. Each segment is labeled using the dimensional affect estimation model of MASOM.

$$\begin{aligned}
 Arousal = & -1.551 & (3.2b) \\
 & + 0.060 * LoudnessMean \\
 & + 0.087 * LoudnessSTD \\
 & + 1.905 * PerceptualTristimulus2STD \\
 & + 0.698 * PerceptualTristimulus3Mean \\
 & + 0.560 * MFCC3STD \\
 & - 0.421 * MFCC5STD \\
 & + 1.164 * MFCC11STD
 \end{aligned}$$

We use the affect estimation model given in Equation 3.2a and 3.2b within two sub-modules: offline labeling of automatic corpus generation in the learning module and online machine listening in the generation module. For example, Figure 3.2 exemplifies the affective labels of an audio corpus. We labeled each audio segment using the equation 3.2a and 3.2b. We explain the details of the audio segmentation in the following section.

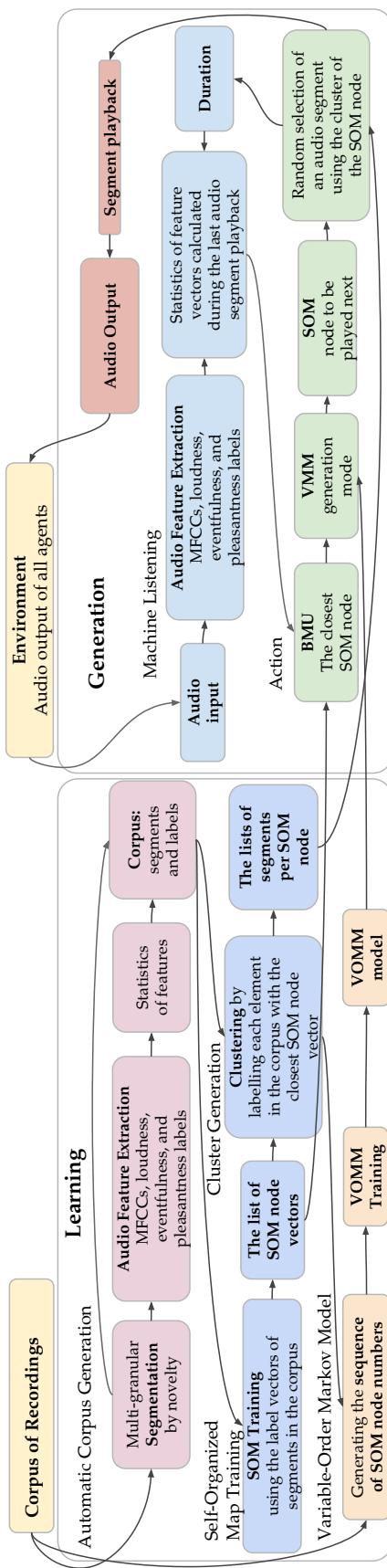


Figure 3.3: The architecture of MASOM

3.4.2 The Learning in MASOM

Automatic Corpus Generation

MASOM automatically generates its memory using an audio corpus. There are two steps of creating the audio samples in the memory: *segmentation* and *labeling* (Figure 3.3).

We segment an audio file using the MIRToolbox³ library in MATLAB. MASOM applies a new segmentation algorithm based on novelty, called *Multi-granular approach* [16]. The MIRToolbox includes this new segmentation algorithm. This approach calculates a novelty curve using the similarity matrix that displays the musical structures in an audio file. The generation of the similarity matrix has two steps. First, the algorithm generates a dissimilarity matrix by calculating the distances between an audio frame and all previous frames. The choice of the type of distance measure relies on the audio feature in focus. Cosine and Euclidean are common distance measures in the similarity matrix calculations. Second, the algorithm converts a dissimilarity to a similarity matrix using one of two equations: linear, $y = 1 - x$ or exponential, $y = \exp(-x)$. To generate the similarity matrix, MASOM uses Fast Fourier Transform (FFT) with 50ms windows with no overlapping to calculate the spectrum. Then, MASOM calculates cosine distances between a frame and its preceding frames to generate the dissimilarity matrix. Lastly, the agent uses the linear conversion to convert the dissimilarity matrix to the similarity matrix.

Using the similarity matrix, the segmentation algorithm calculates of a novelty curve. A novelty curve is the probability of transitions between successive sound objects. The local maxima in novelty curves indicate a high probability of transitions, and therefore, segmentation points. The segmentation procedure ends by saving each segment as a different file to generate an audio corpus of MASOM’s musical memory. Figure 3.4 shows an example of MASOM’s segmentation. In our experiments with a variety of corpus, we observe that this segmentation algorithm successfully creates audio segments ranging from a fraction of a second to several seconds, which corresponds to the sound object time scale.

Following, we label each audio segment with a vector with 31 dimensions. The average and standard deviation of the audio features are computed over the whole segment. Sixteen dimensions are the average of 2 affect estimation features, 13 MFCCs, and loudness. Fourteen dimensions are the standard deviations of 13 MFCCs and loudness. The remaining dimension is the length of the audio segment in seconds. These labels are later used as feature vectors to train the SOM.

The Self-Organizing Map

We use the *ml.som* MAX object to implement the SOM. The object is publicly available by Smith and Garnett [32]. Input vectors of the SOM are 31-dimensional vectors of audio segments in the

³<https://www.jyu.fi/hum/laitokset/musiikki/en/research/coe/materials/mirtoolbox>

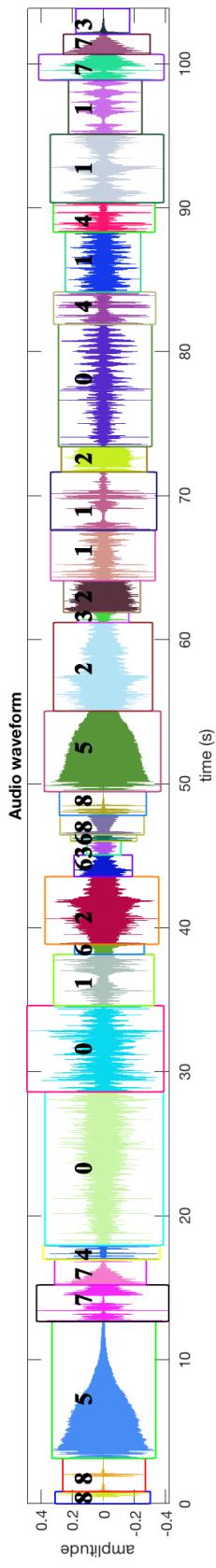


Figure 3.4: The waveform is the seventh track of Bernard Parmegiani's *De Natura Sonorum* album. Each square indicates a novel segment. The corresponding SOM node of the segment is written inside the box.

corpus. The topology of the SOM is rectangular. The size of the topology changes with the number of audio files in the corpus. The SOM topology is $a * a$, where

$$a = \text{int}(\sqrt{\text{the number of audio samples in the memory}/6}) \quad (3.3)$$

Hence, the total number of SOM nodes is approximately one-sixth of the total number of audio files. The total number of epochs is 400 in the training of the SOM. The learning rate is initially set to 0.25 and linearly decreases to 0.001 as the epoch step increases. The neighborhood function is linear, that is, the amount of update is linearly decreasing as the neighboring node is further away from the BMU. The neighborhood radius is initially set to $r = a/2$ and linearly decreases to $r = 1$ as the epoch step increases. We came up with the Equation 3.3 and the parameters of SOMs after several trials with corpora.

As a result of the SOM training, MASOM uses the SOM to generate clusters in the corpus. MASOM labels each audio segment with its BMU. Hence, each SOM node represents a cluster of audio segments. Some nodes may end up with no audio segments after clustering. In our trials, we found the ratio of 6 in equation 3.3 by aiming for the least number of SOM nodes without any audio segments. We further discuss this in the Section Evaluation and Future Work and mention our future work to develop MASOM's memory further.

The Variable-Order Markov Model

MASOM trains a VMM using a string of SOM nodes. First, MASOM labels each audio segment with its BMU in the trained SOM. Second, MASOM uses the original order of the audio segments to create a string of SOM nodes. Then, MASOM trains the VMM using this string.

For example, Figure 3.4 shows the waveform of the seventh track of Bernard Parmegiani's *De Natura Sonorum* album. The track is a minute and forty-three seconds long. Although this track is too short to train MASOM, we exemplify MASOM's training using such a short track. First, MASOM creates novel segments of the track. Each block in Figure 3.4 represents one segment. MASOM found 32 novel segments in this track. Second, MASOM labels each audio segment with a 31-dimensional vector. Third, MASOM sets the topology of the SOM as 3x3 using the equation 3.3. Fourth, MASOM trains the SOM using the label vectors of audio segments. Fifth, MASOM labels each audio segment with its BMU. The BMUs of audio segments in this track is written on each audio segment in Figure 3.4. Using the original order of the audio segments, we create a string of SOM nodes. For this track, the SOM node string is,

$$88577400162636885232112041411773 \quad (3.4)$$

Lastly, MASOM uses this string to train VMM.

We implement VMM in MAX using the VMM java external for MAX ⁴. This external is an implementation of Prediction by Partial Match-Method C (PPM-C) algorithm. PPM-C requires a pre-set maximum Markov order. In MASOM’s architecture, the maximum Markov order is 10. This is because we designed VMM to work in meso time scale of music. Our segmentation algorithm that we explain in the Section Automatic Corpus Generation generates segments ranging from a fraction of a second to several seconds, which is sound object time scale. Hence, a maximum of 10 segments would range from seconds to minutes, which corresponds to the *meso* time scale.

Following, we explain the details of the PPM-C algorithm. The performance of a compression algorithm is tested by calculating the log-loss over a test sequence. Lower log-loss over a test sequence implies better compression rates [3]. However, the probability of an unobserved sequence is zero. Hence, the log-loss of such sequence is infinite. This is known as the zero frequency problem in pattern matching and recognition. PPM-C handles zero frequency using the escape method.

The escape method is as follows. Given a training sequence with length D , for each context s with length $k \leq D$, PPM partitions a total probability, $P_k(\text{escape}|s)$ between all symbols that does not appear after the context s . PPM allocates the remaining probability, $1 - P_k(\text{escape}|s)$ between the symbols that appear after the context s . The PPM variant is determined by how $P_k(\text{escape}|s)$ is calculated and how $1 - P_k(\text{escape}|s)$ is distributed amongst the symbols with non-zero counts [3].

In particular, the escape mechanism of PPM-C is as follows. Given a maximum VMM order n , context s , and symbol σ ,

$$P(\sigma|s) = \frac{f_\sigma}{L + M} \quad (3.5a)$$

$$P(\text{escape}|s) = \frac{M}{L + M} \quad (3.5b)$$

where M is the number of unique symbols in the alphabet, L is the sum of frequency counts of all symbols in s , and f_σ is the frequency count of a symbol σ . When the escape mechanism happens, PPM-C decreases the order by 1 and calculates the probabilities for the order $n - 1$.

We choose the PPM-C in MASOM’s implementation because PPM-C and Decomposed Context Tree Weighting (DE-CTW) are shown to outperform other compression algorithms that are Binary-CTW, Lempel-Ziv 78 (LZ78), improved LZ78 (LZ-MS), and Probabilistic Suffix Trees on predicting MIDI sequences of well-known classical and jazz pieces [3]. We further discuss this in the Section Evaluation and Future Work.

⁴VMM java external for MAX is available at <http://www.am-process.org/main/?portfolio=vmm>

3.4.3 The Generation in MASOM

MASOM’s generation module includes two submodules: online machine listening and musical action, as depicted in Figure 3.3. The environment of the agent is the summation of the audio output of all agents, software or human. MASOM perceives the current musical state of the environment using its musical memory and the online machine listening.

The machine listening module implements feature extraction, affect estimation, and calculation of statistics. The affect estimation algorithm in the generation module is the online version of the algorithm that we explain in the Section Sound Affect Estimation. MASOM calculates the statistics of audio features within the duration of the sample played by the musical action module. When the action module triggers a new sample, the machine listening module clears all statistics. The machine listening module outputs a 31-dimensional vector to the action module.

The musical action module calculates the distances between the vector provided by the machine listening module and the SOM nodes of the agent to find the BMU. This BMU represents the current musical state that is perceived by MASOM. Then, the action module sends the BMU to VMM. VMM keeps track of the history the BMUs to create a context s . Using this context, VMM predicts a SOM node to be played next. We clarify in the Section The Self-Organizing Map that each SOM node represents a cluster of audio segments. When the previous sample playback finishes, MASOM uses the SOM node predicted by VMM to decide on a cluster of audio segments. Lastly, MASOM randomly chooses a sample within the cluster node to generate the audio output.

3.5 Evaluation and Future Work

Examples of MASOM’s output are available online⁵. The online content includes a MASOM trained on Parmegiani’s *De Natura Sonorum* album. We also provide a recording in which audio segments of this album are played randomly. For now, we let our readers decide the success of MASOM by comparing the random playback of audio segments to MASOM’s output in self-listening mode. The online content also includes documentation of MASOM’s public performances with human performers and other MASOM agents. Moreover, the first step of our future work is running evaluation experiments.

As of May 2017, MASOM has performed in various venues in Vancouver, Canada; and Ístanbul, Turkey. The early performances of MASOM were free improvisation in noise music, in a duo setting with the first author. For the first public concert in October 2016, MASOM was trained on a noise album of the first author. The first author commented that MASOM was successful at copying the musical style of the album. The first author also found that playing in a duo setting with the agent was more satisfying than playing solo and the agent was successful at proposing new musical ideas when the human performer ran out of improvisational ideas. Moreover, the first author emphasized

⁵<http://metacreation.net/masom/>

that a stereo performance setting in which the agent was on one channel whereas the human performer was on the other, improved the clarification of the communication between the agent and the human performer. After this public concert, some audience members commented that they would like to see a visualization of the agent. In our future work, we plan to visualize MASOM.

For the second concert in December 2016, we pushed MASOM’s capabilities with a concert with the NOW Ensemble. The ensemble includes saxophone, trumpet, piano, double bass, and drums. NOW Ensemble plays experimental music, free improvisation, and structured improvisation. MASOM was trained of a recording of NOW Ensemble for this second concert. This performance was a challenge because the agent listening was through a microphone instead of a line signal. The acoustic instruments were not amplified and the distance of the instruments affected what the agent was listening. Such setting requires the mixing of acoustic instruments so that the agent listens to a balanced mix of all instruments. In our future work, we plan to study automatic mixing for the machine listening of musical agents.

The third performance of MASOM was again free improvisation in noise music. This performance was in İstanbul in December 2016. The performance had three sections. The first section was a duo of the first author and MASOM. The second section was a duo of two different MASOM agents. The third section was a trio of two MASOM agents and the first author. Some of the audiences informally reported that they preferred the second section without the human performer to the other sections. This three-section piece was also performed in Vancouver, Canada in March 2017.

The fifth performance of MASOM was again in Vancouver in April 2017. The context of this performance was structured improvisation in electro-acoustic music, including MASOM, the first author, and the second author as the performers. The performance had three different sections in which different MASOM agents were playing. The agents were separately trained on Bernard Parmegiani, David Tudor, and Ryoji Ikeda. The studio session of the rehearsal of this performance is available online. Comparing two different takes with Ryoji Ikeda corpus, we recognize that the agent played louder and noisier when the human performers played louder and noisier overall.

The machine learning model that we use for affect estimation in sound is trained on a corpus of soundscape recordings [11]. Although our experiments with this affect estimation model are convincing, we want to develop a new machine learning model using a corpus of experimental electronic music excerpts. We are in the process of designing an empirical evaluation experiment in which participants rank experimental electronic music excerpts on a continuous 2D affective grid. Using such data, we aim to develop a new model to estimate affect of sounds used in experimental electronic music.

In this version of MASOM, the SOM is MASOM’s symbolic memory. There is no hierarchy in SOM, and the topology is static. There is an improved version of SOM, called Growing Hierarchical Self-Organizing Map (GHSOM) [27]. GHSOM introduces hierarchy to SOMs. GHSOM topology is dynamic, and the topology grows with new input data. GHSOM also addresses the problem of SOM nodes with no audio segments that we mention in the Section The Self-Organizing Map. Although GHSOM does not fit Gabora’s second creative memory property, we think that musical

agents can go beyond human capabilities. With MASOM, we want to move towards the notion of musical agents that listen to music more than a human could [5]. GHSOM can help to develop musical agents that can be trained on big data of music.

There are many variants of VMM algorithms. Begleiter et al. [3] present a comparison of the performance of VMM algorithms on text, molecular biology, and music. The authors show that the performance of a VMM algorithm is context-dependent. For example, LZ-MS performs the best on protein classification whereas LZ-MS and LZ78 perform the worst on predicting English text and symbolic representation of music. Within our knowledge, the comparison of the performance of VMM algorithms on predicting patterns of high-level musical states is still to be done. We plan to compare a set of VMM algorithms in MASOM’s system design as a future work.

Bibliography

- [1] Gérard Assayag and Shlomo Dubnov. Using Factor Oracles for Machine Improvisation. *Soft Computing*, 8(9):604–610, August 2004. ISSN 1432-7643, 1433-7479. doi: 10.1007/s00500-004-0385-4. URL <http://link.springer.com.proxy.lib.sfu.ca/article/10.1007/s00500-004-0385-4>.
- [2] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. Omax brothers: a dynamic topology of agents for improvisation learning. In *Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 125–132. ACM, 2006.
- [3] Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004. URL <http://www.jair.org/papers/paper1491.html>.
- [4] Benjamin David Robert Bogart and Philippe Pasquier. Context machines: A series of situated and self-organizing artworks. *Leonardo*, 46(2):114–122, 2013. URL http://www.mitpressjournals.org/doi/abs/10.1162/LEON_a_00525.
- [5] Nick Collins. Towards Machine Musicians Who Have Listened to More Music Than Us: Audio Database-Led Algorithmic Criticism for Automatic Composition and Live Concert Systems. *Computers in Entertainment*, 14(3):1–14, January 2017. ISSN 15443574. doi: 10.1145/2967510. URL <http://dl.acm.org/citation.cfm?doid=3023312.2967510>.
- [6] Shlomo Dubnov, Gerard Assayag, and Arshia Cont. Audio Oracle: A New Algorithm for Fast Learning of Audio Structures. ICMA, 2007. URL <https://hal.inria.fr/hal-00839072/document>.
- [7] T. Eerola and J. K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, January 2011. ISSN 0305-7356, 1741-3087. doi: 10.1177/0305735610362821. URL <http://pom.sagepub.com/cgi/doi/10.1177/0305735610362821>.

- [8] Tuomas Eerola and Jonna K. Vuoskoski. A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli. *Music Perception: An Interdisciplinary Journal*, 30(3):307–340, February 2013. ISSN 07307829, 15338312. doi: 10.1525/mp.2012.30.3.307. URL <http://mp.ucpress.edu/cgi/doi/10.1525/mp.2012.30.3.307>.
- [9] Arne Eigenfeldt and Philippe Pasquier. Real-Time Timbral Organisation: Selecting samples based upon similarity. *Organised Sound*, 15(02):159–166, August 2010. ISSN 1469-8153. doi: 10.1017/S1355771810000154. URL http://journals.cambridge.org/article_S1355771810000154.
- [10] Paul Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3):169–200, May 1992. ISSN 0269-9931. doi: 10.1080/02699939208411068. URL <http://www.informaworld.com/openurl?genre=article&doi=10.1080/02699939208411068&magic=crossref|D404A21C5BB053405B1A640AFFD44AE3>.
- [11] Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society*, 64(9):646–653, September 2016. ISSN 15494950. doi: 10.17743/jaes.2016.0044. URL <http://www.aes.org/e-lib/browse.cfm?elib=18373>.
- [12] Ohad Fried, Zen Jin, and Reid Oda. AudioQuilt: 2d Arrangements of Audio Samples using Metric Learning and Kernelized Sorting. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths University of London, 2014. ISBN 978-1-906897-29-1 (Online). URL <http://www.nime2014.org/technical-programme/proceedings/>.
- [13] Liane Gabora. Cognitive mechanisms underlying the creative process. In *Proceedings of the 4th conference on Creativity & cognition*, pages 126–133. ACM, 2002. URL <http://dl.acm.org/citation.cfm?id=581730>.
- [14] Andrew Hawryshkewich, Philippe Pasquier, and Arne Eigenfeldt. Beatback: A Real-time Interactive Percussion System for Rhythmic Practise and Exploration. *Proceedings of the tenth International Conference on New Interfaces for Musical Expression*, pages 100–105, 2010. URL <http://www.nime.org/>.
- [15] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1–3):1–6, November 1998. ISSN 0925-2312. doi: 10.1016/S0925-2312(98)00030-7. URL <http://www.sciencedirect.com/science/article/pii/S0925231298000307>.
- [16] Olivier Lartillot, Donato Cereghetti, Kim Eliard, and Didier Grandjean. A simple, high-yield method for assessing structural novelty. In *Proceedings of the 3rd International Conference*

- on Music & Emotion (ICME3)), 2013. URL <https://jyx.jyu.fi/dspace/handle/123456789/41611>.*
- [17] Benjamin Lévy, Georges Bloch, and Gérard Assayag. OMaxist dialectics. In *New Interfaces for Musical Expression*, pages 137–140, 2012. URL <https://hal.archives-ouvertes.fr/hal-00706662/>.
 - [18] Steven R. Livingstone, Ralf Mühlberger, Andrew R. Brown, and Andrew Loch. Controlling musical emotionality: an affective computational architecture for influencing musical emotions. *Digital Creativity*, 18(1):43–53, March 2007. ISSN 1462-6268, 1744-3806. doi: 10.1080/14626260701253606. URL <http://www.tandfonline.com/doi/abs/10.1080/14626260701253606>.
 - [19] Brian C. J. Moore, Brian R. Glasberg, and Thomas Baer. A Model for the Prediction of Thresholds, Loudness, and Partial Loudness. *Journal of Audio Engineering Society*, 45(4):224–240, 1997. URL <http://www.aes.org/e-lib/browse.cfm?elib=10272>.
 - [20] Jérôme Nika and Marc Chemillier. Improtek: integrating harmonic controls into improvisation in the filiation of OMax. In *International Computer Music Conference (ICMC)*, pages 180–187, 2012. URL <https://hal.archives-ouvertes.fr/hal-01059330/>.
 - [21] Jérôme Nika, Dimitri Bouche, Jean Bresson, Marc Chemillier, and Gérard Assayag. Guided improvisation as dynamic calls to an offline model. In *Sound and Music Computing (SMC)*, Maynooth, Ireland, July 2015. URL <https://hal.archives-ouvertes.fr/hal-01184642>.
 - [22] Andrew Ortony and Terence J. Turner. What's basic about basic emotions? *Psychological review*, 97(3):315, 1990. URL <http://psycnet.apa.org/journals/rev/97/3/315/>.
 - [23] François Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003. URL <http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.3.333.16861>.
 - [24] Jaak Panksepp. *Affective Neuroscience: The Foundations of Human and Animal Emotions*. Oxford University Press, September 1998. ISBN 978-0-19-802567-2.
 - [25] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An Introduction to Musical Metacreation. *Computers in Entertainment*, 14(2):1–14, January 2017. ISSN 15443574. doi: 10.1145/2930672. URL <http://dl.acm.org/citation.cfm?doid=3023311.2930672>.
 - [26] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, 2004.

- [27] A. Rauber, D. Merkl, and M. Dittenbach. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, 13(6):1331–1341, November 2002. ISSN 1045-9227. doi: 10.1109/TNN.2002.804221.
- [28] Curtis Roads. *Microsound*. The MIT Press, Cambridge, Mass., August 2004. ISBN 978-0-262-68154-4.
- [29] Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine learning*, 25(2-3):117–149, 1996. URL <http://link.springer.com/article/10.1023/A:1026490906255>.
- [30] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, December 1980. ISSN 0022-3514. doi: 10.1037/h0077714. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1981-25062-001&site=ehost-live>.
- [31] Benjamin D. Smith and W. Scott Deal. ML.* Machine Learning Library as a Musical Partner in the Computer-Acoustic Composition Flight. In *the Proceedings of the Joint Conference ICMC14-SMC14*, volume 2014, Athens, Greece, 2014. URL <http://www.smc-conference.net/smc-icmc-2014/images/proceedings/OS19-B09-ML.pdf>.
- [32] Benjamin D. Smith and Guy E. Garnett. Unsupervised Play: Machine Learning Toolkit for Max. In *the Proceedings of International Conference on New Interfaces for Musical Expression 2012*, 2012. URL http://www.nime.org/proceedings/2012/nime2012_68.pdf.
- [33] Greg Surges and Shlomo Dubnov. Feature selection and composition using PyOracle. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013. URL http://www.pucktronix.com/media/papers/Surges_Dubnov_MuME2013_final.pdf.
- [34] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. URL http://acsweb.ucsd.edu/~chw160/pdf/mume2014_vmo_Wang_Dubnov.pdf.
- [35] Cheng-I Wang, Jennifer Hsu, and Shlomo Dubnov. Machine Improvisation with Variable Markov Oracle: Toward Guided and Structured Improvisation. *Computers in Entertainment*, 14(3):1–18, January 2017. ISSN 15443574. doi: 10.1145/2905371. URL <http://dl.acm.org/citation.cfm?doid=3023312.2905371>.

- [36] Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, June 2009. ISBN 9780470519462.
- [37] Marcel Zentner, Didier Grandjean, and Klaus R. Scherer. Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion*, 8(4): 494–521, August 2008. ISSN 1528-3542. doi: 10.1037/1528-3542.8.4.494. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=2008-09984-007&site=ehost-live>.

Chapter 4

A Comparison of Statistical Sequence Models in Musical Agents based on Self-Organizing Maps

KIVANÇ TATAR

JEFF ENS

JONAS KRAASCH

JIANYU FAN

PHILIPPE PASQUIER

Abstract

Musical agents are artificial agents automatizing musical tasks. We specifically focus on a family of audio-based musical agent architectures called Musical Agents based on Self-Organizing Maps (MASOM). MASOM inherits the latest algorithms in machine listening and learns from a large corpus music recordings. The learning is unsupervised, and the agent is flexible to learn from any set of audio recordings. The machine listening algorithm extracts high-level features such as eventfulness, pleasantness, as well as timbral qualities to analyze and model the musical forms. MASOM combines Self-Organizing Maps (SOMs) with statistical sequence models. The SOM organizes encountered sound objects in a latent sonic space whereas the statistical sequence model captures the musical form. In this study, our analysis focuses on the statistical sequence modeling part of the architecture. To simplify our case study, we concentrate on purely generative applications on MASOM where the agent generates without any interaction with other agents. We compare Factor Oracle, Recurrent Neural Networks, and two variations of Variable Markov Models on musical structure modelling in latent sonic space. The analytical study examines these models in terms of n-gram cloning, repetition percentage analysis, and the longest sub-sequence cloning.

Keywords: Musical agents; Multi-Agent Systems; Artificial Intelligence; Musical Metacreation; Computational Creativity; Virtual Reality; Interactive Arts; Contemporary Arts

4.1 Introduction

Musical Agents based on Self-Organizing Maps is a family of agent architectures for interactive music system applications such as autonomous composing, machine improvisation, and audio-based interactive multimedia systems. MASOM architectures apply unsupervised learning on audio recordings; hence, the agent architecture is flexible to be trained on various musical styles. The shared property of MASOM architectures is the combination of the latent sonic space generation algorithm with statistical sequence models. The sonic latent space generation uses an unsupervised, fully-connected neural network algorithm called Self-Organizing Maps (SOMs). SOMs have been applied to various Machine Learning (ML) tasks such as dimensionality reduction, clustering, topology modelling, and latent space generation. While the SOM in MASOM organizes the agent's sound memory in the latent sonic space, the statistical sequence models organize the sound object selections in time. The sequence modelling algorithms are trained on the sound clusters that are generated by the SOM. The sequence modelling algorithms handle the musical temporality and model the musical structure.

We compare four statistical sequence modelling algorithms in MASOM's system architecture: Factor Oracle, Recurrent Neural Networks (RNNs), Variable Markov Model Max-Order variation (VMM-MO), and Variable Markov Model Prediction by Partial Matching Variation C (VMM-PPM-C). These algorithms have been previously used for modelling musical tasks such as composition, assisted composition, continuation, improvisation, melody, harmony, and rhythm generation [3, 28, 27, 25, 43, 8, 4, 19]. Our case is specifically different than the previous applications of these sequence modelling algorithms where the alphabet size is a fixed-number of musical notes. In MASOM's architecture, we train the sequence modelling algorithms on sound cluster indexes that are the SOM node indexes. The total number of SOM nodes varies with the number of sound objects in a corpus. Hence, the alphabet size of sequence modelling algorithms depends on the number of sound objects in MASOM's corpus.

We conduct an analytical analysis of four sequence modelling algorithms for MASOM applications. Our analysis concentrates on two corpora: 1- Electroacoustic music 2- repetitive experimental electronic music. The Electroacoustic corpus has a higher sound object duration variations, while the repetitive experimental music corpus inherits short repetitive sound objects aligned on fixed-grid time structures. In comparison to the previous related works in the literature (see Section Related Work), we focus on mid-size audio corpora. We propose three measures in our analytical analysis: n-gram cloning, repetition percentage analysis, and the longest sub-sequence cloning (Section ??). We give details of how to calculate these measures, and we think that these measures are beneficial to investigate the musical behaviours of statistical sequence models.

4.2 Related Work

OMAX, Improtex, Audio Oracle, Variable Markov Oracle, FILTER, and Mocking-Bird systems exemplify the state of the art of audio-based musical agents [40]. OMAX combines Open Music with Max to create a flexible framework to make musical agent architectures [2, 18]. The statistical sequence model in OMAX is Factor Oracle. Improtex builds on the OMAX by introducing a generation model that is constraint by a pre-defined scenario [25]. The improvisation generation in Improtex applies *anticipation* and *digestion* strategies. The anticipation strategy is searching for a sequence in the memory that matches the agent's input sequence. The digestion strategy predicts the continuation of the matching sequence found in the memory by the digestion strategy.

Audio Oracle [7] and Variable Markov Oracle [42, 44, 43] are possibly the most similar architectures to MASOM. Audio Oracle incorporates the Factor Oracle into the architecture by labeling Factor Oracle states with audio frame clusters. The clustering of Audio Oracle uses a threshold to recognize similar audio frames. The Audio Oracle sets the threshold using a statistical measure from Informatics, called the Information Rate. Audio Oracle applies the audio frame threshold that gives the highest Information Rate to cluster audio frames. The Variable Markov Oracle builds on the Audio Oracle by introducing the probabilistic generation of Variable Markov Models to the Audio Oracle.

FILTER stands for Freely Improvising, Learning and Transforming Evolutionary Recombination [26]. The agent architecture is inspired by the Smalley's gesture and texture categorizations of sound objects in electroacoustic music composition [38]. FILTER inherits an audio buffer in the scale of seconds to encode the temporal changes in the audio buffer. The agent's generation combines Hidden Markov Models with a mutation only Genetic Algorithm (GA) for its adaptive goal decision process.

Mocking-bird combines the CLARION cognitive architecture with the FILTER [20]. The actions of this agent is decided by the cognitive architecture. The CLARION consists of four modules, that are the Action Control System (ACS), Non-Action Control System (NACS), Meta-cognition System (MCS), and motivation system (MS). The four modules decides on a pre-recorded sample to be played starting from a point with a duration, and post processing effects such as pitch shift and time stretch.

Similar to MASOM, these agents that are mentioned above use audio recordings for learning. All these musical agents are trained on the fly during the live performance using an audio buffer in the range of several minutes. In comparison, MASOM pushes the size of the learning corpus towards hours of music recordings. In the following, we explain MASOM's system in detail.

4.3 System Architecture Details

The architecture of MASOM creates a sound memory through automatic audio segmentation and thumbnailing, using audio features of timbre, loudness, fundamental frequency, duration, and sound

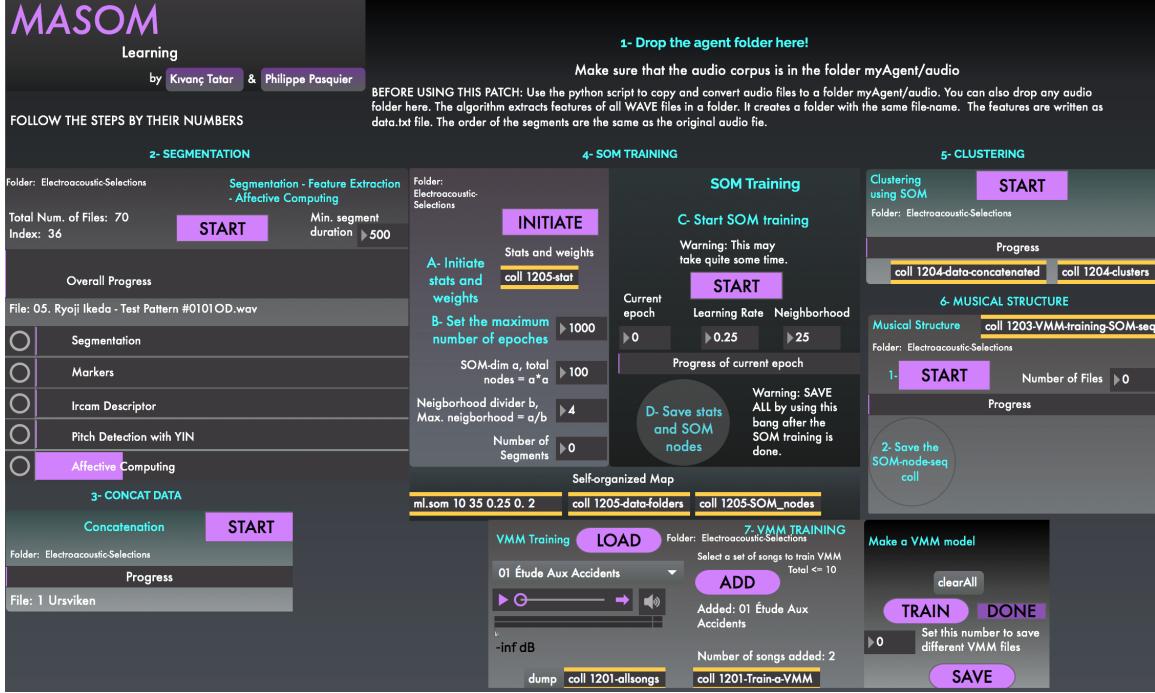


Figure 4.1: The user interface of MASOM’s learning algorithms

affect features of eventfulness and pleasantness. The architecture applies Self-Organizing Maps [15, 16], a neural network machine learning algorithm. Using the SOM, the agent organizes sounds on a two-dimensional map so that similar sound clusters locate closer to each other [10]. MASOM learns the temporality of musical form by applying pattern recognition on the organized sound memory. The agent assumes the musical form as temporal shifts on sound clusters that are organized in the feature space.

4.4 Machine Listening

The machine listening module consists of two steps: segmentation and thumbnailing.

4.4.1 Segmentation

The *segmentation* module in MASOM applies a novelty based onset detection. Müller ([22]) clarifies four types of novelty based onset detection: energy-based, spectrum-based, phase-based, and the complex domain. The energy-based onset detection calculates the difference between the local energy values of two successive window. Then, the signal passes through a half-wave rectification since the onsets are the increase in the energy. The resulting waveform is also referred as the energy-based novelty function. Then, the algorithm applies a peak detection algorithm to find the exact onset locations. There are several peak detection algorithms for audio signals that we can apply [14].

However, in the case of polyphonic audio, a low frequency sound can suppress the detection of an onset that occurs in the high frequency range. To address this masking problem, the spectrum based onset detection calculates the power on each band of the spectrum using FFT. MASOM applies a spectrum-based onset detection. The agent calculates the power of each band and applies ITU-R 468 loudness criteria (1990) to calculate the power of each FFT band. Then, MASOM sums the power on each band and further applies logarithmic scaling to calculate the novelty curve. Then, the agent applies a peak detection algorithm to find exact onset locations. The peak detection algorithm uses a moving median filter and the mean calculation as the onset detection function.

Spectrum-based onset detection algorithms use only the magnitude spectrum whereas phase-based onset detection algorithms utilizes the phase of the complex coefficients of spectrum. There are also complex-domain onset detection functions that use both magnitude and the phase spectrum. We found that the spectrum-based onset detection was satisfactory in the development of MASOM.

We use MAX MuBu¹ external library [37] for the segmentation and the audio feature extraction. The MuBu library includes the IRCAM descriptors audio feature extraction library and Plug-In-Plug-Out (PIPO)² signal processing tools. IRCAM descriptors library includes a vast number of audio features including the perceptual ones. PIPO tools are useful to create a pipeline for feature extraction and signal processing.

4.4.2 Thumbnailing

The hybrid corpus of the agent consists of both audio segments and their thumbnails represented as a 35-dimensional vector. Three types of musical agent corpora have emerged in the musical agent literature: audio, symbolic, and hybrid [40]. Audio corpora consist of recordings whereas symbolic corpora store a representation of music such as MIDI notes, and timbre feature vectors. Hybrid corpora includes both audio recordings and symbolic representations of these recordings.

Audio Feature Set

MASOM automatically labels the audio segments generated by the segmentation module. The thumbnailing module label the segments with a multi-dimensional vector. This vector includes statistics (mean and standard deviation) of audio features. The statistics are calculated per audio segment. The calculation of audio features for segment thumbnailing applies the IRCAM MuBu and PIPO externals. MASOM thumbnails each segment with a 35-dimensional vector of:

- **Timbre features:** mean and standard deviation of 13 MFCCs (26 dimensions) and perceptual spectral decrease (2 dimensions)
- **Loudness:** mean and standard deviation of loudness (2 dimensions)

¹<http://forumnet.ircam.fr/product/mubu-en/>

²<http://ismm.ircam.fr/pipo/>

- **Duration:** the length of segments (1 dimension)
- **Fundamental frequency:** mean and standard deviation of fundamental frequency (2 dimensions)
- **High-level features:** pleasantness and eventfulness (2 dimensions)

In the following, we explain the details of audio features in MASOM’s machine listening:

- **Loudness:** The agent uses the loudness calculation algorithm proposed by [21]. The algorithm pipeline has 6 modules, a filter that models the transfer function of the outer ear, another filter that imitates the transfer function of the middle ear, the calculation of the cochlea excitation pattern, the transformation of the excitation pattern to the equivalent rectangular bandwidths, the calculation of specific loudness (the loudness of each frequency band), the summation of the specific loudness.
- **MFCC:** MASOM applies Mel Frequency Cepstral Coefficients (MFCCs) as the audio feature that represents the timbre [30]. The MFCCs feature has been shown to outperform other audio features in the task of timbre matching using a single-objective fitness function [45]. MFCCs are calculated in three successive steps: spectrum (FFT), the logarithm of the spectrum, Discrete Cosine Transform (DCT) using Mel frequency scale. MASOM calculates 13 MFCCs excluding the zero coefficient. The zero coefficient corresponds to the DC offset; and therefore, it is removed.
- **Perceptual Spectral Decrease:** For a given spectrum at time t , spectral decrease is a measure of average decrease in the spectral amplitude. The perceptual spectral decrease is the spectral decrease computed using a human hearing model [30].
- **Affective features:** The machine listening module of MASOM incorporates an sound affect estimation algorithm. Affective Computing field studies computational models to predict the emotional response to a stimuli. For example, the stimuli can be an image, a video, a human body posture, a sound, or a music piece. Several discrete and continuous affect models for estimating the affective state of a sound have been proposed in the literature [9]. In addition to the two continuous dimensions of valence and arousal, some models include a third dimension of potency, dominance, or tension.

MASOM implements a two dimensional affect model that generates a continuous output in the range $[-1, 1]$. The algorithm applies a continuous two dimensional affect model and outputs a vector with two dimensions: pleasantness and eventfulness. These two features correspond to valence and arousal, respectively. The name of affective dimensions are different in the sound literature because of the following. Russell [36] proposes sixteen emotions that are circularly arranged around the continuous affect dimensions of valence and arousal. However, these emotions are the attributions of the subject, but not the stimuli. That is, there is no such thing as “a miserable sound”, but one can feel miserable after hearing a sound. Therefore, Fan et al. (2016)

proposed pleasantness and eventfulness dimensions to explain the perceived affective dimensions of sound.

MASOM's machine listening module includes two affective features: pleasantness and eventfulness. Fan et al. (2016) shared a sound affect estimation dataset that includes 125 soundscape excepts with 6 seconds duration. This dataset is labeled by 20 participants on the 2-dimensional affective grid. We used this ground-truth dataset to train our affect estimation model and We applied the multivariate linear regression machine learning model. We implemented the model using PIPO MAX externals to have the same model available for both offline thumbnailing and online machine listening.

We use the equations 4.1a and 4.1b to calculate the pleasantness and eventfulness of a sound,

$$\begin{aligned}
 \text{Pleasantness} = & - 0.169 + & (4.1a) \\
 & - 0.061 * \text{LoudnessMean} \\
 & + 0.588 * \text{SpectralFlatness1Mean} \\
 & + 0.302 * \text{MFCC1STD} \\
 & + 0.361 * \text{MFCC5STD} \\
 & - 0.229 * \text{PerceptualSpectralDecreaseSTD}
 \end{aligned}$$

$$\begin{aligned}
 \text{Eventfulness} = & - 1.551 & (4.1b) \\
 & + 0.060 * \text{LoudnessMean} \\
 & + 0.087 * \text{LoudnessSTD} \\
 & + 1.905 * \text{PerceptualTristimulus2STD} \\
 & + 0.698 * \text{PerceptualTristimulus3Mean} \\
 & + 0.560 * \text{MFCC3STD} \\
 & - 0.421 * \text{MFCC5STD} \\
 & + 1.164 * \text{MFCC11STD}
 \end{aligned}$$

In addition the audio features that we use to thumbnail MASOM's segments, the affect estimation machine learning model include the following audio features:

- **Perceptual Tristimulus:** Tristimulus is a feature that is inspired by the properties of color in vision [31]. Tristimulus is a set of energy ratio formulas that describe the harmonics in the spectrum of a sound. There are three different types of tristimulus that corresponds to first, second, and the third harmonics. Perceptual Tristimulus calculation applies a human hearing model to filter the incoming signal before the tristimulus calculation [30].
- **Spectral Flatness:** Spectral Flatness is the ratio of the geometric mean to the arithmetic mean of an energy spectrum, which gives a measure of noisiness vs. sinusoidality of a spectrum. MASOM calculates the spectral flatness for 4 different bands 250 - 500, 500 - 1000, 1000 - 2000, 2000 -

4000 Hz. *SpectralFlatness1Mean* in equation 4.1a refers to average spectral flatness of the band 250-500 Hz.

- **Fundamental Frequency:** Roads ([33]) defines the fundamental frequency of a sound as “the rate of repetition of a periodic sound.” We use YIN algorithm to estimate the fundamental frequency of a sound [5]. The YIN algorithm has been shown to overcome previous approaches by giving lower error rates. The algorithm applies difference function with cumulative mean normalization instead of the autocorrelation function. YIN further improves the fundamental frequency estimation by introducing an absolute threshold, parabolic interpolation to find the abscissa of a local minimum, and best local estimate procedure to decrease the local fluctuations of the estimate frequency

The Design Iterations of the Audio Feature Selection

We previously proposed that the first step of informal evaluations of Musical Metacreation systems is the author evaluations [40]. The audio features included in the label vectors of audio samples are handpicked through a design process where the first and the last authors evaluated MASOM’s musical output and the clustering accuracy of SOMs with several corpora of electroacoustic, IDM, noise, avangard jazz, techno, and house music. We first started with two affect features of eventfulness and pleasantness. We realized that using these two affect features, the system was unable to grasp the timbre information in the recordings although the eventfulness patterns in the recordings appeared in the output of MASOM. As the second step, we added a comprehensive timbre feature set, 13 MFCCs which is shown to outperform other audio features in the task of audio timbre matching [45]. This feature addition greatly improved the success of clustering and the MASOM’s musical output. However, in rare cases, we observed that some clusters included samples that we perceived to fit in two different, but closely related clusters. Hence, in the third step, we included perceptual spectral decrease feature to further improve the SOM clustering. In the three initial iterations of audio feature selections, we mainly focused on IDM and noise music where the loudness variations and the dynamic range was low. Our experiments with electroacoustic corpus indicated that the agent’s musical output was jumping between various loudness levels in a random manner. Hence, in the fourth step, we added the loudness audio feature to the set. In the last step, we added duration of samples to the feature set to decrease duration variations within SOM clusters.

In the following, we delve into clustering the audio segments using the selected audio feature labels.

4.5 Organizing Sounds with Self-Organizing Maps

Self-Organizing Map (SOM) is an artificial neural network (ANN) for dimensionality reduction, clustering, and visualization applications. SOMs apply unsupervised learning to automatically generate a 2-dimensional symbolic representation of data. Common topologies of SOMs are square,

rectangular, and toroid shapes. The map consists of a finite number of nodes. A node is a vector with the same number of dimension as the input data. In SOM, similar nodes locate closer to each other. Therefore, SOMs take the shape of input topology and can show the similarities in the input data.

At each training step, a SOM processes one input vector from the dataset. First, the SOM finds the closest node (n_c) in the map for the given input vector. Then, the SOM updates n_c and a number of neighboring nodes so that they move closer to the input vector. SOM calculates the update amount of n_c using a parameter called the *learning rate*. SOM decides the number of neighboring nodes that move with n_c using a parameter called *neighborhood*. The update amount of neighboring nodes is a percentage of the update amount of n_c . The update amount of a neighboring node decreases as the node is further away from n_c . This decrease depends on the *neighboring function*. Two of common neighboring functions are linear and Gaussian.

The training procedure of a SOM passes the training dataset more than once. Each pass is called an *epoch*. It is common to decrease the learning rate and the neighbouring radius as the epoch number increases. Because of this decrease in learning rate and the neighbouring radius, SOMs can capture the generalised topology as well as the particularities of the training data.

After the training, we can use an SOM to cluster the input data. Each node in a SOM represents one cluster. In the clustering using SOMs, the input vectors belong to the closest SOM node cluster. Depending on the topology of data set, some SOM nodes may not cluster any input vectors, and these SOM nodes stand as empty clusters. The clustering procedure applies a distance measure to find the closest SOM node of an input vector (n_c). Euclidean and cosine distances are common distance measures that are applied to calculate n_c . Euclidean distance calculation takes the magnitudes of vectors into consideration whereas the magnitude of vectors do not affect the cosine distance. Using a trained SOM, we can cluster both the training data or new data.

In MASOM's learning, the topology is a square of $a * a$. We aim for 6 samples per node in average. In our trials, aiming 6 samples per cluster gave a low number of empty clusters. We point out the future directions for clustering in Section Sparsity. MASOM uses the following formula to calculate the parameter a :

$$a = \text{int}(\sqrt{\text{the number of audio samples}/6}) \quad (4.2)$$

The total number of epochs is 1000. The learning rate is linearly decreasing from 0.25 to 0.01 with each epoch. The neighbourhood radius is also linearly decreasing from $a/4$ to 0. For a given audio corpus, we calculate the mean and the standard deviation of each audio feature, and use these statistics to standardize the SOM input vectors:

$$I_{norm}[i] = \frac{I[i]}{M[i]} * STD[i] \quad (4.3)$$

and $i \in [1 : N]$, where the N is the total number of dimensions of the input vector, I is the calculated audio feature vector, M is a vector that gives the average of each audio feature, STD is

a vector of standard deviation of each audio feature, and I_{norm} is the normalized feature vector. M and STD vectors are calculated for a given corpus.

In addition to the normalization, we also apply a weight vector W to balance the effect of each audio feature on the Euclidean distance calculation. The weights of MFCC dimensions are 1/13 given that 13 MFCCs are calculated. The rest of the weight vector dimensions are set to 1. Hence, the total MFCC distances read as a timbre distance, and MFCC distances do not suppress the effect of remaining features on distance calculation. The SOM training input vector is,

$$I_{SOM} = I_{norm} * W \quad (4.4)$$

We have concluded on the equation 4.2, 4.3, and 4.4 after several trials with variety of corpora including electro-acoustic music, noise and glitch, IDM, and mainstream electronic music.

4.6 Musical Structure Generation with Statistical Sequence Modelling Algorithms

After the automatic creation of sonic latent space with SOMs, the question is, how do we model the organization of sound objects in time? MASOM learns the temporal musical structure by using a symbolic representation of compositions. This symbolic representation is a string of SOM nodes, and Figure 4.2 illustrates how the agent generates this string. First, MASOM labels each audio segment in a recording with its closest SOM node in the trained map. Second, MASOM uses the original order of the audio segments to create a string of SOM nodes. This procedure is repeated for each recording. Then, the agent learns the musical structure using these symbolic representations of recordings.

We researched four algorithms for the musical structure modelling in MASOM: Variable-Markov Model Max-order variation (VMM-MO), Variable Markov Model Prediction by Partial Matching - C variation (VMM-PPM-C), Factor Oracle, and Recurrent Neural Networks. In the following, we explain how these algorithms work.

4.6.1 Variable-Markov Models

Markov Models are finite state machines that model patterns in discrete sequences using the Markov assumption. An N^{th} order Markov model assumes the following:

$$P(s_t | s_{t-1}, s_{t-2}, \dots, s_1) = P(s_t | s_{t-N}, \dots, s_{max(t-N,1)}) \quad (4.5)$$

Hence, Markov models are history dependent. The order of a Markov model indicates how many previous states to be considered to predict or generate the next state. Moreover, the frequency of transitions between Markov states determines the conditional probabilities of the transitions. Therefore, a Markov Model is a stochastic model represented as a directional graph.

Variable Markov models (VMMs) refer to a family of algorithms such as probabilistic finite automata, probabilistic suffix automata, prediction suffix trees, Lempel-Ziv 78, improved Lempel-

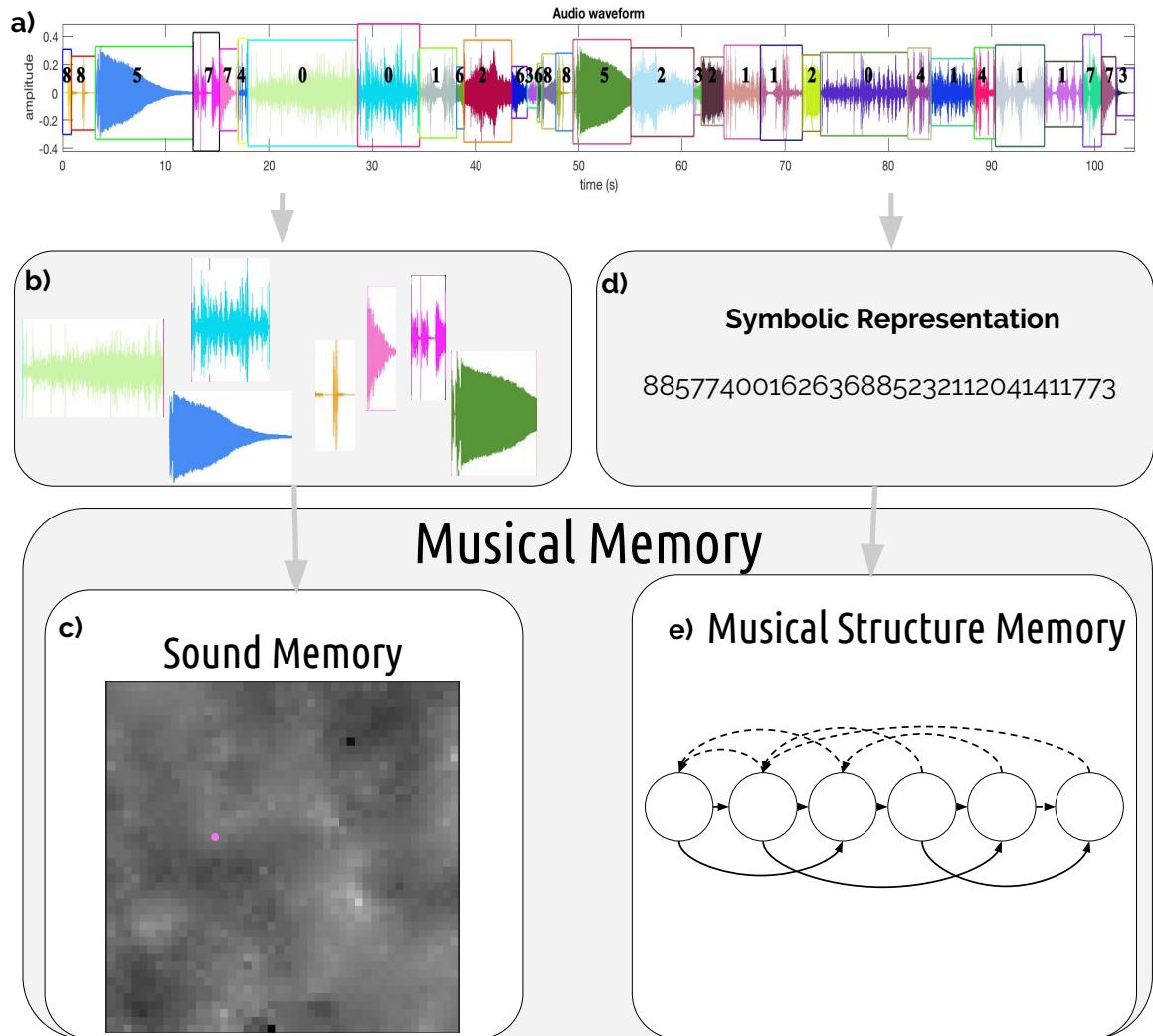


Figure 4.2: The training of MASOM: a) Segmentation b) Labelling the audio samples c) Sound memory where squares stand for SOM nodes that are a clusters of audio samples d) Creating a symbolic representation of the original song using the clusters indexes of audio samples e) statistical sequence model to learn temporal transitions.

Ziv (LZ-MS), Prediction by Partial Match (PPM), Factor Oracles, and the context tree weighting method [34, 3]. VMM considers a varying number of previous states to predict the next state. VMMs are applied to a variety of MUME tasks in the musical agents literature [40, p. 25–30].

The log-loss indicates the performance of a compression algorithm where a lower log-loss implies better compression rates [3]. However, a VMM model does not necessarily observe all possible sequences during the training. The probability of an unobserved sequence is zero. Hence, the log-loss of such sequence is infinite. This is known as the zero frequency problem in pattern matching and recognition. VMM variations handle zero frequency using an escape method.

Max-Order Variation

The escape method of VMM-MO is as follows. When the model encounters a sequence that hasn't been observed during the training, the order decreases by one and the symbol at the most distant past is discarded. This process is repeated until a known sequence is found. The model uses the found known sequence to predict the next symbol. In the most extreme case, the model decreases the order to zero, and predicts a symbol using the empty context. The max-order VMM has been previously shown to be successful for musical applications [28, 27].

We implement a max-order VMM where the model is a list of transition matrices and count matrices stored as nested lists. Each nested list represents the transition matrix of a Markov Model of an order. The implementation is a Java code in MAX using the *max-mxj* framework.

Prediction by Partial Matching-C Variation

The escape method of VMM-PPM introduces a probabilistic approach to the escape method of VMM-MO. Given a training sequence with length D , for each context s with length $k \leq D$, VMM-PPM partitions a total probability, $P_k(\text{escape}|s)$ between all symbols that do not appear after the context s . PPM allocates the remaining probability, $1 - P_k(\text{escape}|s)$ between the symbols that appear after the context s . The PPM variant is determined by how $P_k(\text{escape}|s)$ is calculated and how $1 - P_k(\text{escape}|s)$ is distributed amongst the symbols with non-zero counts [3].

Several variants of VMM-PPM have been proposed in the literature. In particular, the escape mechanism of VMM-PPM-C is as follows. Given a maximum VMM order n , context s , and symbol σ ,

$$P(\sigma|s) = \frac{f_\sigma}{L + M} \quad (4.6a)$$

$$P(\text{escape}|s) = \frac{M}{L + M} \quad (4.6b)$$

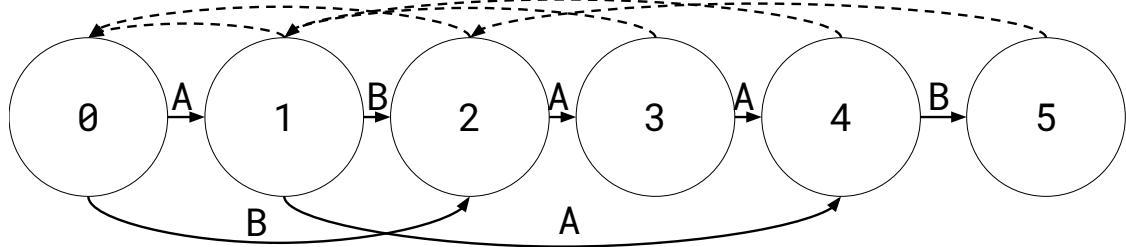


Figure 4.3: Factor Oracle generated using the sequence $ABAAB$.

where M is the number of unique symbols in the alphabet, L is the sum of frequency counts of all symbols in s , and f_σ is the frequency count of a symbol σ . When the escape mechanism happens, PPM-C decreases the order by 1 and calculates the probabilities for the order $n - 1$.

4.6.2 Factor Oracle

Similar to the VMM learning, MASOM variations with Factor Oracle (FO) use the SOM string representation of a recording to train the FO. FO is a finite state automata that represents substrings and patterns in a sequence, that is, at least all *factors* of a sequence (Figure 5.4). FO has three types of links: internal links, external links, and suffix links. Internal links are forward links between successive states. External links are forward links that jump longer than successive states. Suffix links are backward links that point the longest repeating factor in previous states. FO allows incremental learning, and learning is linear in time and space [17]. [1] compare IP, PSTs, and FOs for the symbolic sequences of music. [1] conclude that FOs suit the best to satisfy incremental and fast on-line learning, time-bounded generation of musical sequences, and implementation of multi-attribute models to deal with the multi-dimensionality of music. Within the last two decades, many studies implemented FOs in musical agents [1, 2, 6, 18, 23, 39, 26, 42, 24, 25].

4.6.3 Recurrent Neural Networks

Recurrent Neural Networks (RNNs) [35] are a type of neural network which is characterized by the presence of feedback connections, allowing the hidden activations to flow in a loop. As a result, these networks are adept at modelling time-series data. A basic RNN architecture is defined by the following equations, where x_t denotes the input, \hat{x}_t denotes the output, and u_t denotes the hidden units at time-step t . Therefore, an RNN has five learnable parameters, W_{vu} , W_{uu} , W_{uv} , b_u , and b_v , which can be trained using back propagation through time:

$$u_t = \sigma(W_{vu}x_t + W_{uu}u_{t-1} + b_u) \quad (4.7)$$

$$\hat{x}_t = \sigma(W_{uv}u_t + b_v) \quad (4.8)$$

A Long Short Term Memory Network (LSTM) is a type of RNN, which has additional gates to control the flow of information through time [12]. We used Tensorflow’s Keras API for our RNN implementation using LSTMs as cellblocks ³.

In the following, we explain our evaluation methodology and the results of our analyses.

4.7 Evaluation Methodology

We previously proposed a typology of evaluation methodologies for musical agent architectures [40]. The first layer of this typology consists of two categories: informal and formal evaluations. Informal evaluations do not involve formalized research methodologies. Common subjects of these type of evaluations are the authors of system architectures, users, peers and experts, audience members, and the media. In comparison, formal evaluations employ formalized methodologies to assess the success of musical agent systems. Three categories emerge in formal evaluations: theoretical and analytic measures, empirical studies, and the feedback of peer reviewers, curators, and jury. Theoretical and analytical measures apply computational analysis while empirical studies investigate case studies with human participants. In this study, we employ a theoretical and analytic measures of n-gram cloning, repetition percentage analysis, and the longest sub-sequence cloning for musical structure modeling algorithms.

Given a corpus of sequences $C = \{c_0, \dots, c_k\}$, and corpus of generated sequences $X = \{x_0, \dots, x_l\}$, we apply three metrics to evaluate the quality of X with respect to C . Let $\|x\|$ denote the length of a sequence x , $x_{i,j}$ denote a sub-sequence of X including all elements from i up to j , and $\mathbf{I}(s, S)$ denote a function that returns 1 if the sequence s is found within S , and 0 otherwise.

4.7.1 N-gram Cloning

The fist measure estimates the degree to which the generated material copies sub-sequences from the original data. Cloning has been previously referred as plagiarism and identified as a metric when generating material from a trained model [29]. We measure n-gram cloning using the following formula, which measures the number of sub-sequences of window size n that are found in the original corpus. We compute this measure for $n \in [1, 10]$ to evaluate the level of cloning at different scales.

$$\text{cloning}(x, C, n) = \frac{\sum_{i=0}^{\|x\|-n+1} \mathbf{I}(x_{i,i+n}, C)}{\|x\| - n + 1} \quad (4.9)$$

The range of n-gram cloning is $[1, 0]$ where 1 means every single windowed n-gram is present in the training corpus and no unique n-gram is found in the material generated by the statistical sequence model.

³<https://www.tensorflow.org/guide/keras>

4.7.2 Repetition Percentage Analysis

The second measure estimates the variance of sub-sequences within each generated sequence. This measure can be used to analyze the level of repetition within a sequence, as well as comparison of repetition levels in the generated sequences and the sequences in the corpus. We expect a statistical sequence model to situate in between the two extremes of a substantial amount of repetition and the absence of any repetition. We measure repetition using the following formula, where $x_{0,i}$ denotes the entire sub-sequence preceding $x_{i,i+n}$.

$$\text{repetition}(x, C, n) = \frac{\sum_{i=0}^{\|x\|-n+1} \mathbf{I}(x_{i,i+n}, x_{0,i})}{\|x\| - n + 1} \quad (4.10)$$

A high value of repetition indicates that sub-sequences are often repeated and there is minimal variance in the generated sequence. In contrast, a low value of repetition indicates a high degree of variance as few sub-sequences are ever repeated.

4.7.3 The Longest Sub-sequence Cloning

The third measure is simply the longest sub-sequence which is found in both the generated sequence (x) and the original data (C). We propose the length of the longest subsequence as another measure for understanding the level cloning of a statistical sequence model. The training of a model aims for a balance where the model captures the patterns in the sequence, and avoids generating extremely long sequences that are the exact copies of the sub-sequences in the learning data.

4.8 Results

We first give details of the corpora that we used for this study. We continue by clarifying the details of sequence generation cases, and then we analyze the generated results.

4.8.1 Corpora

Our analyses focus on two mid-size corpora: 1- electroacoustic music, 2- repetitive experimental music. The electroacoustic music corpus inherits well-known acousmatic compositions of twentieth century experimental music composers. The corpus spans more than 8 hours of music, including 57 recordings of well-known composers. The total number of sound segments of this corpus is 11899 and total number of SOM nodes is 1936. Figure 4.4 visualizes the dimensions of the SOM trained on the Electroacoustic music corpus.

The repetitive experimental electronic music corpus has examples of IDM and noise music with repetition. This corpus has 93 recordings spanning almost 7 hours of music. The number of sound segments in this corpus is 42276 and the number of total SOM nodes is 7744. Figure 4.5 shows the dimensions of the SOM trained on the IDM corpus. By comparing Figure 4.4-35 and 4.5-35, we can

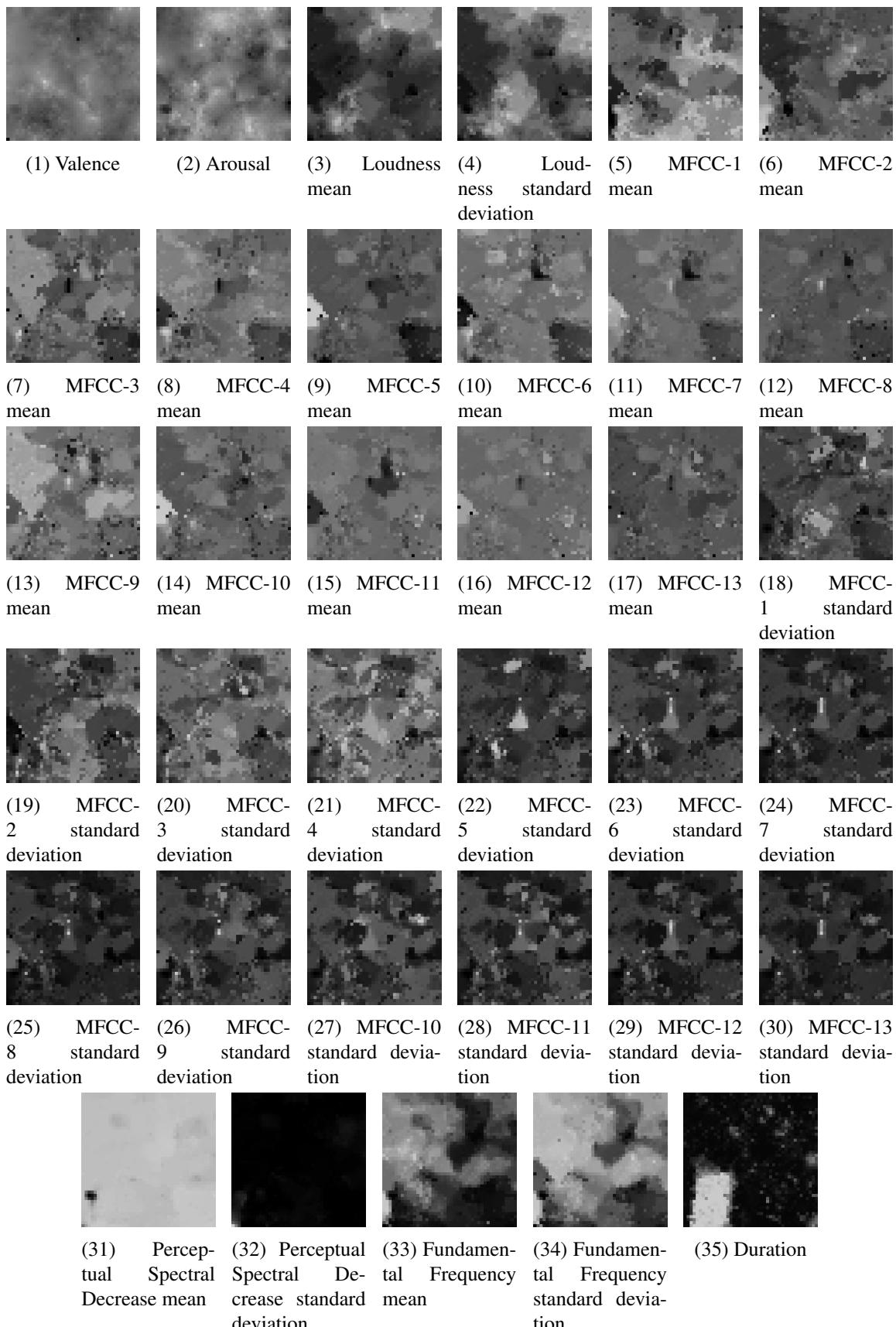


Figure 4.4: The dimensions of SOM trained on the audio segments of Electroacoustic music corpus. 152
Each figure shows one audio feature listed in Section 4.4.2. The black and white represent the minimum and the maximum values, respectively.

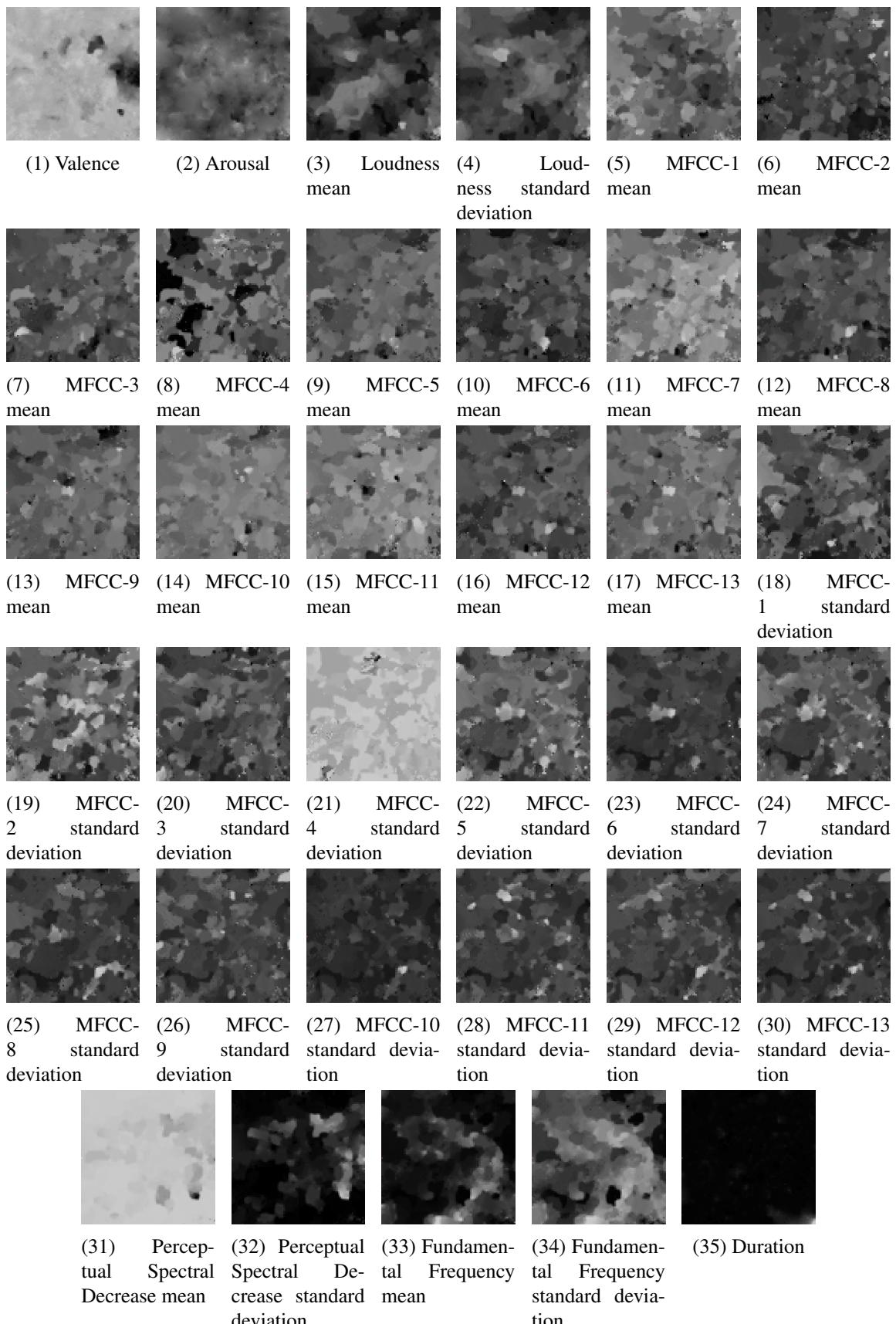


Figure 4.5: The dimensions of SOM trained on the audio segments of repetitive experimental electronic music corpus. Each figure shows one audio feature listed in Section 4.4.2. The black and white represent the minimum and the maximum values, respectively.

observe that the electroacoustic music corpus has a higher variation of duration than the repetitive experimental music corpus.

4.8.2 Generated Sequences

For each corpus, we have the following cases:

- *CORPUS_ALL*: We trained the statistical sequence models on all sequences of SOM node indexes corresponding to all recordings in the corpus.
- *CORPUS_[idx]*: The statistical sequence models are trained on the sequences of SOM indexes associated to a subset of recordings in the corpus.

The *CORPUS_[idx]* cases are trained on approximately five stylistically consistent songs selected from the corpus, including all sequence modelling algorithms that are RNN, Factor Oracle, 3rd order VMM-PPM-C, 5th order VMM-PPM-C, 3rd order VMM-MO, and 5th order VMM-MO. In comparison, the *CORPUS_ALL* case excludes VMM-MO models because VMM-MO implementation uses the random-access memory and the model runs out of space for the *CORPUS_ALL* case. For all cases, we also added the *RANDOM* case where the sequences are generated randomly. We elaborate on the aesthetic reasoning of choosing a subset of songs for training sequence models in Section 4.9.

For each statistical sequence model and *CORPUS_[idx]* combination, we generated 10000 sequences with the length 2048 symbols. In the case of *CORPUS_ALL*, we generated 10000 sequences with the length 12001 symbols, for each statistical sequence model. The analyses graphs shown in Figure 4.6, 4.7, 4.8, 4.9, 4.10 4.11 use the generated sequences mentioned above⁴.

4.8.3 Analyses

We applied three analysis metrics of n-gram cloning, the longest sub-sequence cloning and repetition percentage on sequences generated by statistical sequence models. The first metric, n-gram cloning analysis shows how much unique n-grams are generated by the model, where 1 means every single windowed n-gram generated by the model is present in the training corpus. Our analysis of n-gram cloning for all cases in Figures 4.6 and 4.7 show that all models resulted in n-gram cloning percentages that were less than 6% for all n-grams with two exceptions. In the first exception, the fifth-order VMM-MO model produced 20% n-gram cloning percentage for the second-order in the case of *CORPUS_2* in Figure 4.6. In the second exception, the Factor Oracle outputted 22.4% n-gram cloning for the second-order in the case of *CORPUS_ALL*.

Figures 4.8 and 4.9 present the repetition percentage analysis on the statistical sequence models. The third and fifth order of VMM-MO models generated the highest percentage of repetition

⁴Example audio files are available at <https://kivanctatar.com/masom-1-30>.

whereas the third and fifth order VMM-PPM-C, and RNN generated the least percentage of repetition with the random generation. Note that, Figures 4.8 and 4.9 also show that VMM-MO models and Factor Oracle introduce more repetitions than the level of repetitions in the training corpus. The results of repetition analysis were consistent across corpora.

The third metric that we focus on is the longest sub-sequence cloning. Figures 4.10 and 4.11 are the bar-plots of this metric. We notice that the longest sub-sequence cloning lengths were lower than 10 for all cases. All variations of Variable Markov Models could not copy sequences longer than the model order. In the case of the Factor Oracle, we observe that the models copied longer sequences for the *CORPUS_ALL* cases compared to the remaining cases. We observe that the results of the longest sub-sequence cloning were varying across corpora.

4.9 Discussions

Our evaluation methodology includes two measures of cloning percentage. Previously, the term cloning has been referred as plagiarism in the context of analyzing statistical sequence models [29]. We prefer to exchange the term plagiarism with cloning because plagiarism suggests a negative connotation. We think that there could be artistic applications where cloning is acceptable. For example, artists can use audio-based musical agent architectures with unsupervised learning to convert their fixed-media recordings to generative or interactive music systems. In that case, the artist is the author of the content that the agent learns. Artists may prefer the generative output of agents to be strictly consistent to their style.

4.9.1 How much cloning is acceptable?

We hope that our analyses provide a descriptive understanding of how much cloning appears when a particular statistical sequence model is chosen. Across statistical sequence models and corpora cases, the n-gram cloning percentages were substantially low. This brings us to the question, how low is too low? A reference level for this measure for the audio applications with mid-size and big-size corpora is currently not available. In our future work, we plan to expand the number of corpora cases and study a reference level for this measure for comparison of machine learning algorithms for time-series sequence generations.

If we take a step back from the statistical analysis and look from the musical perspective, we should consider that the amount of acceptable cloning may depend on the musical style. For example, the kick-snare drum pattern with the hi-hat on the offbeat, is very common in the House music genre of mainstream electronic music. This type of cloning is perceived as a stylistic feature in comparison to plagiarism. Another example where cloning is embedded as a stylistic feature is *Mashup* where the music is produced mainly by manipulating samples of other music recordings. We can also give a contrasting musical genre as an example where almost no cloning appears. In electroacoustic music, composers rarely copy material from other composer's work. Hence, the acceptable amount of cloning is in relation with the distinct properties of the musical style.

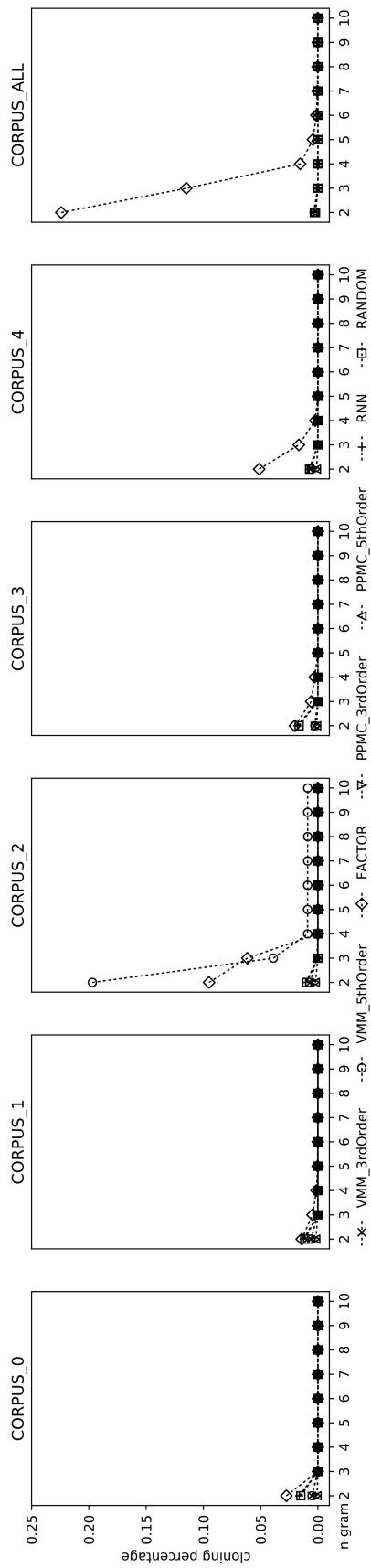


Figure 4.6: The percentage of n-gram cloning in the generated sequences using the corpora of electroacoustic music.

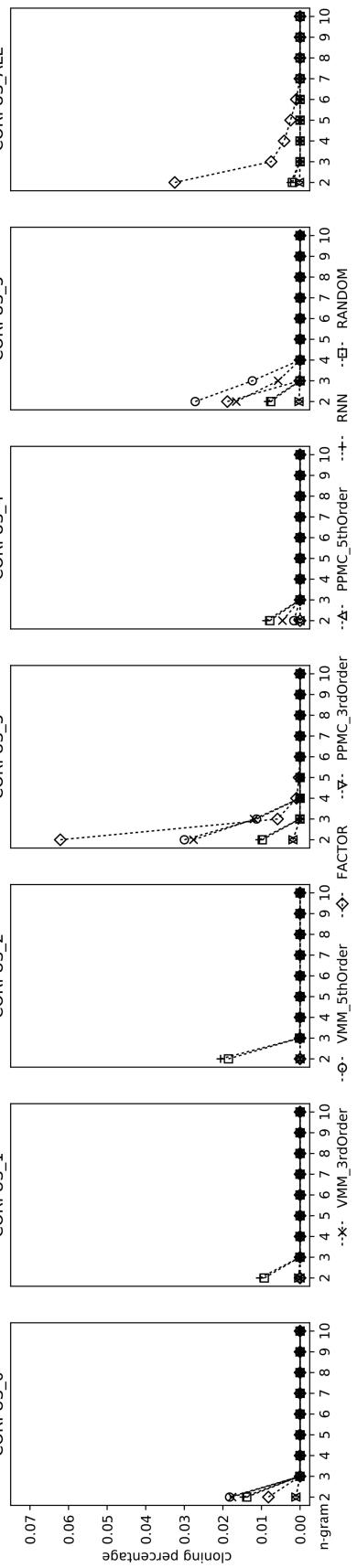


Figure 4.7: The percentage of n-gram cloning in the generated sequences using the corpora of repetitive experimental music.

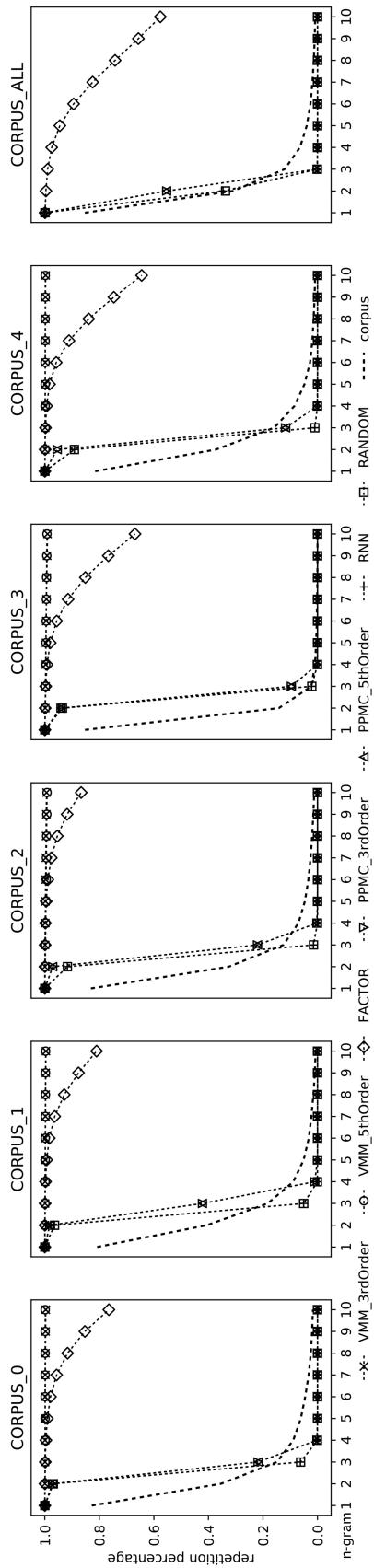


Figure 4.8: The percentage of repetition for Electroacoustic music corpus

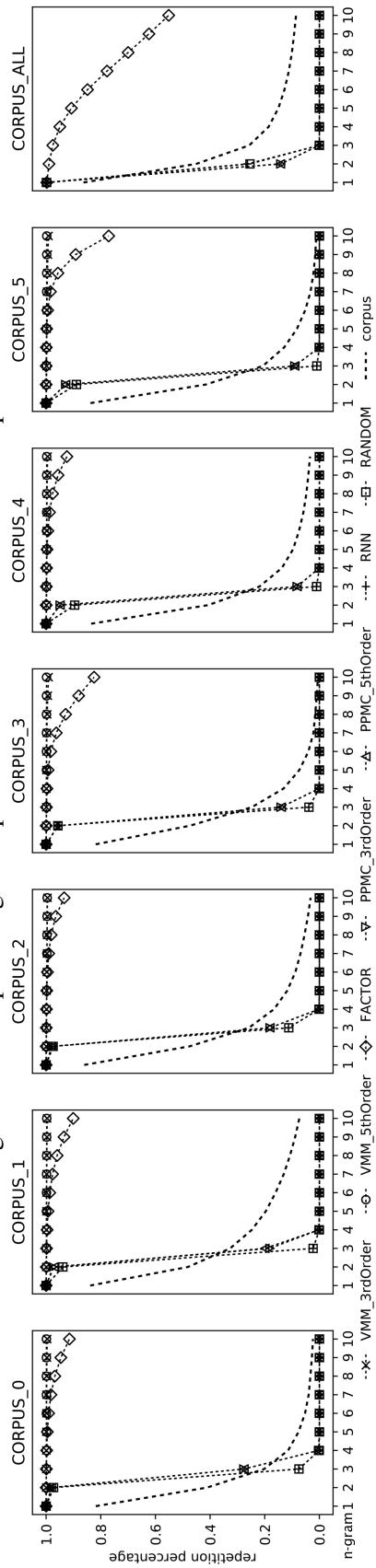


Figure 4.9: The percentage of repetition for IDM music corpus

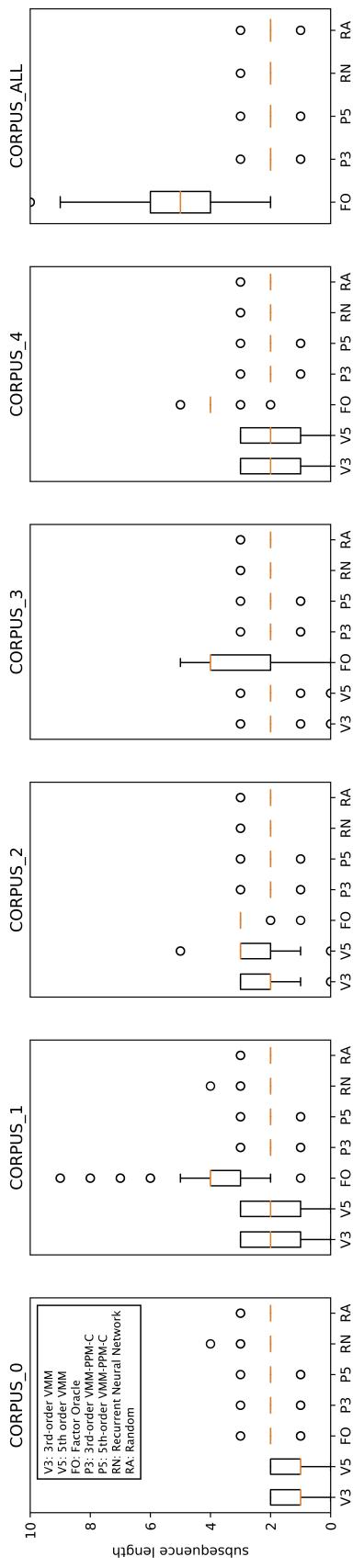


Figure 4.10: The box-plot of the longest subsequence cloning in the generated sequences are calculated using the corpora of electroacoustic music. The red line represent the mean, while the circles indicate the outliers.

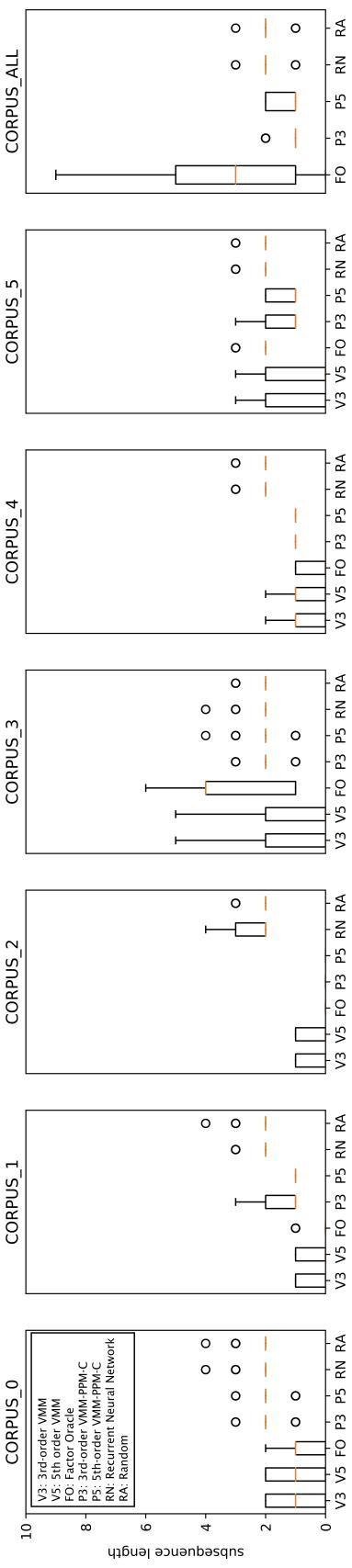


Figure 4.11: These statistics of the longest subsequence cloning in the generated sequences are computed using the corpora of IDM music. The red lines represent the mean, while the circles indicate the outliers.

4.9.2 How much repetition is preferable?

Some music styles involves a high level of repetition, such as mainstream electronic music and IDM. In comparison, electroacoustic music often avoids the repetition of sound objects on a fixed time grid. We think that the amount of preferred repetition depends on the specific musical applications. In general, our repetition analysis shows that the outputs of Factor Oracle models resulted in a balanced percentage of repetition. However, the extremes of repetition percentage metric can still be useful for specific artistic applications. For example, artists can use the VMM-MO models to generate a higher level of repetition than the repetition in the training corpus for musical styles with repetitive sound objects.

4.9.3 Sparsity

As we mentioned in Section 4.8.3, n-gram cloning percentages were lower than 6% and the lengths of the longest sub-sequence cloning were lower than 10. The longest sub-sequence cloning lengths for VMM models were lower than the order for all cases. This fact made us to question the sparsity of the audio clustering with the corpora. Our analysis indicates that the clustering procedure excelled in specificity; however, the model lost generality that the time-series models require for the learning.

The dichotomy of generality versus specificity frequently appears in machine learning applications. For example, the dichotomy of generality versus specificity has been addressed in Evolutionary Computation for sound synthesizer preset generation applications using two strategies: exploration and exploitation of the search space [41]. In the applications of MASOM, we observed that SOMs successfully captured the specificity of the latent audio space. The clustering could capture the subtle variations between sound objects. However, the single layer architecture of plain SOM does not provide generality of the latent space. To address this issue, Growing-Hierarchical Self-Organizing Maps (GHSOM) have been proposed in the machine learning literature [32]. In Figure 4.4 and 4.5, we can observe clusters of clusters across different dimensions. The hierarchical learning in GHSOM can improve both generality and specificity of the clustering. This type of latent audio space generation combined with a hierarchical time-series learning could be an exciting opportunity for a musical agent architecture.

4.9.4 Curse of Dimensionality

The difference between the level of repetitions in the training dataset and ML models in Figures 4.10 and 4.11 intrigued us to analyze the success of audio representation using SOMs. Although we observed that the audio clustering using SOMs combined with MASOM's audio feature set gives perceptually consistent clusters⁵, this does not necessarily indicate that the 2D topology of SOM is the best representation to train statistical sequence models. The number of clusters generated by SOM increases the alphabet size of statistical sequence models (see Section 4.8.1), and this may

⁵A video that exemplifies SOM audio clusters can be found at <https://kivanctatar.com/masom-1-30>

make it harder for sequence models to capture the temporal patterns. We think that a hierarchical clustering model such as Growing Hierarchical Self-Organizing Maps (GHSOM) [32] combined with a hierarchical and statistical sequence model can address the alphabet size issue. The tree data structures generated by GHSOM may capture granularity of subtle audio variations while keeping the generality required to train the statistical sequence models.

4.10 Conclusion

Musical Agents based on Self-Organizing Maps (MASOM) is a family of agent architectures for electronic music, mainly focusing on experimental music applications. The agent architectures enable artist to use fixed musical recordings to create musical agents for generative and interactive music implementations. In this paper, we explained the technical details of MASOM architecture. The training includes machine listening, sound object memory generation, and musical structure modelling. The machine listening algorithm applies onset detection for segmentation, and thumbnailling to label sound objects. These labels are later used by the Self-Organizing Maps training for sound object memory generation. The generated SOM is the latent sonic space where similar sounds locate closer to each other. The musical structure generation algorithm puts the generated latent sonic space into practice to train statistical sequence modelling algorithms. We focus on four sequence modelling algorithms: Variable Markov Model Max-Order variation, Variable Markov Model Prediction by Partial Matching C variation, Recurrent Neural Networks, and Factor Oracle. MASOM's training of sequence models is different than the previous applications of sequence modelling algorithms because the alphabet size of sequences depends on the total number of SOM nodes. The number of SOM nodes increases with the number of sound objects in the training corpus (Equation 3.3). Hence, the size of alphabet changes with the size of number of sound objects in the corpus.

Our analyses of four statistical sequence algorithms applied three metrics of n-gram cloning, repetition percentage analysis, and the longest sub-sequence cloning. We propose that these three metrics can provide descriptive understanding of sequence modeling algorithms for musical applications. We provided the details of how to calculate these metrics so that the musical agent literature can benefit analytical measures to compare musical outputs of agent systems. Our case study focused on two corpora, one with higher degree of repetition (repetitive experimental electronic music corpus) than the other (Electroacoustic music corpus). The results of three metrics gave an idea of the musical behaviors of trained statistical sequence models for our particular application.

Bibliography

- [1] G. Assayag and S. Dubnov. Using Factor Oracles for Machine Improvisation. *Soft Computing*, 8(9):604–610, August 2004. ISSN 1432-7643, 1433-7479. doi: 10.1007/s00500-004-

- 0385-4. URL <http://link.springer.com.proxy.lib.sfu.ca/article/10.1007/s00500-004-0385-4>.
- [2] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. Omax brothers: a dynamic topology of agents for improvisation learning. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 125–132. ACM Press, 2006. URL <http://dl.acm.org/citation.cfm?id=1178742>.
- [3] Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004. URL <http://www.jair.org/papers/paper1491.html>.
- [4] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, December 2014. URL <http://arxiv.org/abs/1412.3555>. arXiv: 1412.3555.
- [5] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917, 2002. ISSN 00014966. doi: 10.1121/1.1458024. URL <http://scitation.aip.org/content/asa/journal/jasa/111/4/10.1121/1.1458024>.
- [6] Shlomo Dubnov, Gerard Assayag, and Arshia Cont. Audio Oracle: A New Algorithm for Fast Learning of Audio Structures. In *Proceedings of International Computer Music Conference*, 2007. URL <https://hal.inria.fr/hal-00839072/document>.
- [7] Shlomo Dubnov, G. Assayag, and A. Cont. Audio Oracle Analysis of Musical Information Rate. In *Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC)*, pages 567–571, September 2011. doi: 10.1109/ICSC.2011.106.
- [8] Douglas Eck and Jurgen Schmidhuber. A First Look at Music Composition using LSTM Recurrent Neural Networks. *Istituto Dalle Molle Di Studi Sull’Intelligenza Artificiale*, 103:11, 2002.
- [9] Tuomas Eerola and Jonna K. Vuoskoski. A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli. *Music Perception: An Interdisciplinary Journal*, 30(3):307–340, February 2013. ISSN 07307829, 15338312. doi: 10.1525/mp.2012.30.3.307. URL <http://mp.ucpress.edu/cgi/doi/10.1525/mp.2012.30.3.307>.
- [10] Arne Eigenfeldt and Philippe Pasquier. Real-Time Timbral Organisation: Selecting samples based upon similarity. *Organised Sound*, 15(02):159–166, August 2010. ISSN 1469-8153. doi: 10.1017/S1355771810000154. URL http://journals.cambridge.org/article_S1355771810000154.

- [11] Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society*, 64(9):646–653, September 2016. ISSN 15494950. doi: 10.17743/jaes.2016.0044. URL <http://www.aes.org/e-lib/browse.cfm?elib=18373>.
- [12] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- [13] International Telecommunication Union. Recommendation ITU-R BS.468: Measurement of audio-frequency noise voltage level in sound broadcasting. In *ITU-R recommendations: BS series; Broadcasting service (sound)*. International Telecommunication Union, Geneva, 1990.
- [14] Ismo Kauppinen. Methods for detecting impulsive noise in speech and audio signals. In *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, volume 2, pages 967–970. IEEE, 2002. URL <http://ieeexplore.ieee.org/abstract/document/1028251/>.
- [15] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982. URL <http://link.springer.com/article/10.1007/BF00337288>.
- [16] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1–3):1–6, November 1998. ISSN 0925-2312. doi: 10.1016/S0925-2312(98)00030-7. URL <http://www.sciencedirect.com/science/article/pii/S0925231298000307>.
- [17] Arnaud Lefebvre and Thierry Lecroq. A Heuristic For Computing Repeats With A Factor Oracle: Application To Biological Sequences. *International Journal of Computer Mathematics*, 79(12):1303–1315, January 2002. ISSN 0020-7160, 1029-0265. doi: 10.1080/00207160214653. URL <http://www.tandfonline.com/doi/abs/10.1080/00207160214653>.
- [18] Benjamin Lévy, Georges Bloch, and Gérard Assayag. OMaxist dialectics. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 137–140, 2012. URL <https://hal.archives-ouvertes.fr/hal-00706662/>.
- [19] Zachary C. Lipton, John Berkowitz, and Charles Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv:1506.00019 [cs]*, May 2015. URL <http://arxiv.org/abs/1506.00019>. arXiv: 1506.00019.
- [20] Michael F. Lynch. Motivation, Microdrives and Microgoals in Mockingbird. In *Proceedings of 3rd International Workshop on Musical Metacreation (MUME 2014)*, North Carolina, USA, 2014. URL <http://musicalmetacreation.org/mume2014/content/proceedings/Motivation,%20Microdrives%20and%20Microgoals%20in%20Mockingbird.pdf>.

- [21] Brian C. J. Moore, Brian R. Glasberg, and Thomas Baer. A Model for the Prediction of Thresholds, Loudness, and Partial Loudness. *Journal of Audio Engineering Society*, 45(4):224–240, 1997. URL <http://www.aes.org/e-lib/browse.cfm?elib=10272>.
- [22] Meinard Müller. *Fundamentals of Music Processing*. Springer International Publishing, Cham, 2015. ISBN 978-3-319-21944-8 978-3-319-21945-5. URL <http://link.springer.com/10.1007/978-3-319-21945-5>.
- [23] Jérôme Nika and Marc Chemillier. Improtex: integrating harmonic controls into improvisation in the filiation of OMax. In *International Computer Music Conference (ICMC)*, pages 180–187, 2012. URL <https://hal.archives-ouvertes.fr/hal-01059330/>.
- [24] Jérôme Nika, Dimitri Bouche, Jean Bresson, Marc Chemillier, and Gérard Assayag. Guided improvisation as dynamic calls to an offline model. In *Sound and Music Computing (SMC)*, Maynooth, Ireland, July 2015. URL <https://hal.archives-ouvertes.fr/hal-01184642>.
- [25] Jérôme Nika, Marc Chemillier, and Gérard Assayag. ImprotexK: Introducing Scenarios into Human-Computer Music Improvisation. *Computers in Entertainment*, 14(2):1–27, January 2017. ISSN 15443574. doi: 10.1145/3022635. URL <http://dl.acm.org/citation.cfm?doid=3023311.3022635>.
- [26] Doug Van Nort, Pauline Oliveros, and Jonas Braasch. Electro/Acoustic Improvisation and Deeply Listening Machines. *Journal of New Music Research*, 42(4):303–324, December 2013. ISSN 0929-8215. doi: 10.1080/09298215.2013.860465. URL <http://dx.doi.org/10.1080/09298215.2013.860465>.
- [27] François Pachet. Beyond the cybernetic jam fantasy: The continuator. *Computer Graphics and Applications, IEEE*, 24(1):31–35, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1255806.
- [28] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003. URL <http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.3.333.16861>.
- [29] Alexandre Papadopoulos, Pierre Roy, and François Pachet. Avoiding plagiarism in markov sequence generation. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI’14, pages 2731–2737. AAAI Press, 2014. URL <http://dl.acm.org/citation.cfm?id=2892753.2892930>.
- [30] Geoffroy Peeters, Bruno L. Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

- [31] H. F. Pollard and E. V. Jansson. A Tristimulus Method for the Specification of Musical Timbre. *Acta Acustica united with Acustica*, 51(3):162–171, August 1982.
- [32] A. Rauber, D. Merkl, and M. Dittenbach. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, 13(6):1331–1341, November 2002. ISSN 1045-9227. doi: 10.1109/TNN.2002.804221.
- [33] Curtis Roads. *Composing electronic music: a new aesthetic*. Oxford University Press, Oxford, 2015. ISBN 978-0-19-537324-0.
- [34] Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine learning*, 25(2-3):117–149, 1996. URL <http://link.springer.com/article/10.1023/A:1026490906255>.
- [35] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, Cambridge, MA, USA, 1988. ISBN 0-262-01097-6. URL <http://dl.acm.org/citation.cfm?id=65669.104451>.
- [36] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, December 1980. ISSN 0022-3514. doi: 10.1037/h0077714. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1981-25062-001&site=ehost-live>.
- [37] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Riccardo Borghesi. MuBu and friends—assembling tools for content based real-time interactive audio processing in Max/MSP. In *Proceedings of International Computer Music Conference (ICMC)*, 2009.
- [38] Denis Smalley. Spectromorphology: explaining sound-shapes. *Organised Sound*, 2(02):107–126, August 1997. ISSN 1469-8153. doi: 10.1017/S1355771897009059. URL http://journals.cambridge.org/article_S1355771897009059.
- [39] Greg Surges and Shlomo Dubnov. Feature selection and composition using PyOracle. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013. URL http://www.pucktronix.com/media/papers/Surges_Dubnov_MuME2013_final.pdf.
- [40] Kivanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):1–50, September 2018. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1511736. URL <https://www.tandfonline.com/doi/full/10.1080/09298215.2018.1511736>.

- [41] Kivanç Tatar, Matthieu Macret, and Philippe Pasquier. Automatic Synthesizer Preset Generation with PresetGen. *Journal of New Music Research*, 45(2):124–144, April 2016. ISSN 0929-8215. doi: 10.1080/09298215.2016.1175481. URL <http://dx.doi.org/10.1080/09298215.2016.1175481>.
- [42] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. URL http://acsweb.ucsd.edu/~chw160/pdf/mume2014_vmo_Wang_Dubnov.pdf.
- [43] Cheng-i Wang and Shlomo Dubnov. Context-Aware Hidden Markov Models of Jazz Music with Variable Markov Oracle. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, 2017.
- [44] Cheng-i Wang, Jennifer Hsu, and Shlomo Dubnov. Machine Improvisation with Variable Markov Oracle: Toward Guided and Structured Improvisation. *Computers in Entertainment*, 14(3):1–18, January 2017. ISSN 15443574. doi: 10.1145/2905371. URL <http://dl.acm.org/citation.cfm?doid=3023312.2905371>.
- [45] Matthew John Yee-King. *Automatic sound synthesizer programming: techniques and applications*. PhD thesis, University of Sussex, 2011. URL <http://core.ac.uk/download/pdf/2710683.pdf>.

Chapter 5

Audio-based Musical Artificial Intelligence and Audio-Reactive Visual Agents in Revive

KIVANÇ TATAR

PHILIPPE PASQUIER

REMY SIU

IN PROCEEDINGS OF THE INTERNATIONAL COMPUTER MUSIC CONFERENCE 2019,
JUNE 2019

Abstract

Revive is a live audio-visual performance project that brings together a musical artificial intelligence architecture, human electronic musicians, and audio-reactive visual agents in a complex multimedia environment of a dome view with multichannel 3D audio. The context of the project is live audio-visual performance of experimental electronic music through structured improvisation. *Revive* applies structured improvisation using cues and automatized parameter changes within these cues. Performers have different roles within the musical structures initiated by the cues. These roles change as the performance temporally evolves. Sonic actions of performers are further emphasized by audio-reactive visual agents. The behaviours and contents of sonic and visual agents change as the performance unfolds.

Keywords: Musical agents; Multi-Agent Systems; Artificial Intelligence; Musical Metacreation; Computational Creativity; Virtual Reality; Interactive Arts; Contemporary Arts

5.1 Introduction

Revive is a live audio-visual performance project that features two human performers, and MASOM, which is a musical agent, an artificial intelligence (AI) architecture for live performance¹. For each sonic performer in *Revive*, a corresponding visual agent puts sonic gestures and textures into live generative images. The visual agents use a machine listening algorithm in the input module to exhibit audio-reactive behaviours with generative visuals. This reveals the musical gestures that are so often lost in electronic music performance.

Revive's aesthetics span a variety of experimental electronic music styles including acousmatic music, soundscapes, glitch, intelligent dance music (IDM), and noise music. Acousmatic compositions use electronic means to create or process sounds to produce compositions. Soundscapes use field recordings of the environment outside the studio as the musical content. Glitch music explores the idea of using sounds that are generated by the failure of any procedure. For example, glitch performers overload the cpu to generate clicks and drops on the audio output. IDM composers use any sound object to produce dance music, extending their audio palette to unconventional sounds. Glitch sounds such as clicks, short impulsive noises frequently appear in IDM compositions. Noise music stands on the louder and aggressive end of the musical composition continuum. Noise music employs loud sounds to stimulate the body. The stimulations can be an ear pain caused by the loud sounds or pulsations generated by loud bass frequencies to vibrate the human body.

The musical AI in *Revive*, Musical Agent based on Self-Organizing Maps (MASOM) is an audio-based musical agent with autonomous unsupervised learning. Musical agents are artificial agents that automatize musical creative tasks [19]. Musical agents differ from purely generative systems because of autonomy, reactivity, proactivity, adaptability, coordination, and emergence behaviours.

The architecture of MASOM proposes an innovative approach by combining a sonic latent space generation with statistical sequence modelling for temporal musical structure. The autonomous, unsupervised learning in MASOM only requires audio recordings. The musical agent creates a sound memory through automatic audio segmentation and thumbnailing, using audio features of timbre, loudness, fundamental frequency, duration, and music emotion features of eventfulness and pleasantness. The agent organizes sounds on a two-dimensional map so that similar sound clusters locate closer to each other. MASOM learns the temporality of musical form by applying statistical sequence modelling on the organized sound memory. Hence, the agent assumes the musical form as temporal shifts on sound clusters that are organized in the feature space. In *Revive*, we use a variation of MASOM architecture, where previous statistical sequence modelling algorithm, VMM-PPM-C is exchanged with the Factor Oracle algorithm [2] to improve timbre consistency for the cases where the agent is trained on big-size audio recording data ranging from 1-GB to 100GB.

¹A recording of a *Revive* session is available at <https://kivanctatar.com/revive> where the audio is the binaural encoding of the 3D audio setup.

Revive project exemplifies how to incorporate an unsupervised, audio-based musical AI system into an audio-visual live performance. The cue system of *Revive* automatizes parameter changes that initiate musical sections in the performance. This allows performers to focus more on the aesthetics and less on the technical complexities. The cue system defines roles where the performers can explore improvisation within musical roles. Some examples of these roles are filling the background sonic canvas, generating repetitive bass with fast spatial movements, improvising within a constrained spectrum, improvisation through reaction towards the sonic gestures of musical AI, and staying quiet. Hence, the cues automatically set a structured improvisation setup in the complex performance environment of *Revive*. In the context of improvised (or non-idiomatic) music, structured improvisation is free improvisation with predefined constraints for musical sections. In addition, the synchronization of 3D audio spatialization with the audio-reactive generative visuals emphasizes the sonic gestures of performers in *Revive*. The specifics of 3D audio setup that we mention in this paper clarify the technical details of the performances at the Société des Arts Technologique (SAT) dome with 157 speakers clustered to 31 audio channels². However, the spatialization tools in *Revive* is flexible for any 3D or 2D audio speaker setup.

In the following, we first give a brief introduction to the Creative Artificial Intelligence and Multi-agents Systems. We continue by explaining the performance setup of *Revive*. Then, we delve into the reactive agent architecture of the visual agents. We move further with the technical details and aesthetic background of sonic strategies in *Revive*. Finally, we conclude by the discussions around previously mentioned topics while proposing possible future steps.

5.2 Multi-agent Systems in Creative Artificial Intelligence for Music and Multimedia

Creative Artificial Intelligence (AI) for Music explores the applications of autonomous systems of Applied AI and Multi-agent Systems (MAS) for musical applications [14]. Autonomous system architectures for creative tasks differ from conventional architectures that aim to solve tasks with optimal solutions. Creative tasks often lack optimality, and the quality measures are ill-defined. For example, there is no universal objective measure to assess if one acousmatic composition is better than any other. The lack of objective measures asks for system designs where the possibilities of connections between autonomous behaviours and artistic aesthetics are explored. In many cases, the architectures work with a set of hyper-parameters where the user can manipulate the autonomous behaviour by exploring the space of these parameters.

Artificial agents in MAS are autonomous software with perception and action capabilities. Musical agents are implementations of Multi-agent systems combined with applied artificial intelligence and machine learning algorithms for musical applications. We previously clarified six levels of mu-

²<http://sat.qc.ca/en/satosphere>

sical agent behaviours: 1- Reactivity, 2-Proactivity, 3- Interactivity, 4-Adaptability, 5-Versatility, 6-Volition and framing; where the higher levels can inherit properties of the lower levels [19].

The visual agent architecture in *Revive* aims for reactive behaviours whereas the musical AI, MASOM's initial architecture with VMM-PPM-C can exhibit interactive and adaptive behaviours. In *Revive*, we give up on the adaptive behaviours of MASOM with the Factor Oracle variant (MASOM-FO) to improve timbre consistency with mid-size datasets. The interactivity of MASOM-FO is more on a higher level through user interaction, where the user can change the statistical sequence model on the fly.

5.3 The Performance Setup

Three sonic performers (including the musical AI MASOM) improvise in *Revive* and the performers are constrained within certain musical roles that are defined per musical sections. The performers apply various sonic strategies that we clarify in Section 5.5. The sonic actions of performers are visualized with a visual agent with audio-reactive behaviours. The localization of generated visuals follows the 3D spatial location of the audio. In addition to 3D audio with three sources, the second author also outputs a background channel that is directly send to all speakers. This helps to create a sonic background canvas for the performers in the foreground.

The *Revive* performance applies a cue system to handle parameter changes automatically (Figure 5.1). The cues also define musical sections where the sonic performers improvise within certain roles. The cues are automatically initiated at certain moments using a timeline. The cue system is implemented within the Jamoma³ framework in Max⁴ [12]. This framework automatically gathers all parameters of Max abstractions coded as Jamoma modules. The Jamoma's cue system allows a simple scripting where the parameters can be linearly ramped to any value. The script engine can also put into a halt for a certain time using the "WAIT" command. These features provide a simple, yet powerful coding of complex performance environments where many parameters constantly change. The cue system with Jamoma, sonic strategies in *Revive*, and machine listening modules of visual agents are implemented in Max. The visuals are generated and rendered to the dome view in Derivative's Touch Designer⁵. These two applications are networked through an OSC communication.

5.4 Audio-reactive visual agents in *Revive*

The visual agents aim to improve the audience's perception of sonic gestures. The visual agents are generative and reacts to the sonic performers by using a machine listening algorithm. Using reactive

³<http://jamoma.org>

⁴<https://cycling74.com/products/max>

⁵<https://www.derivative.ca/>

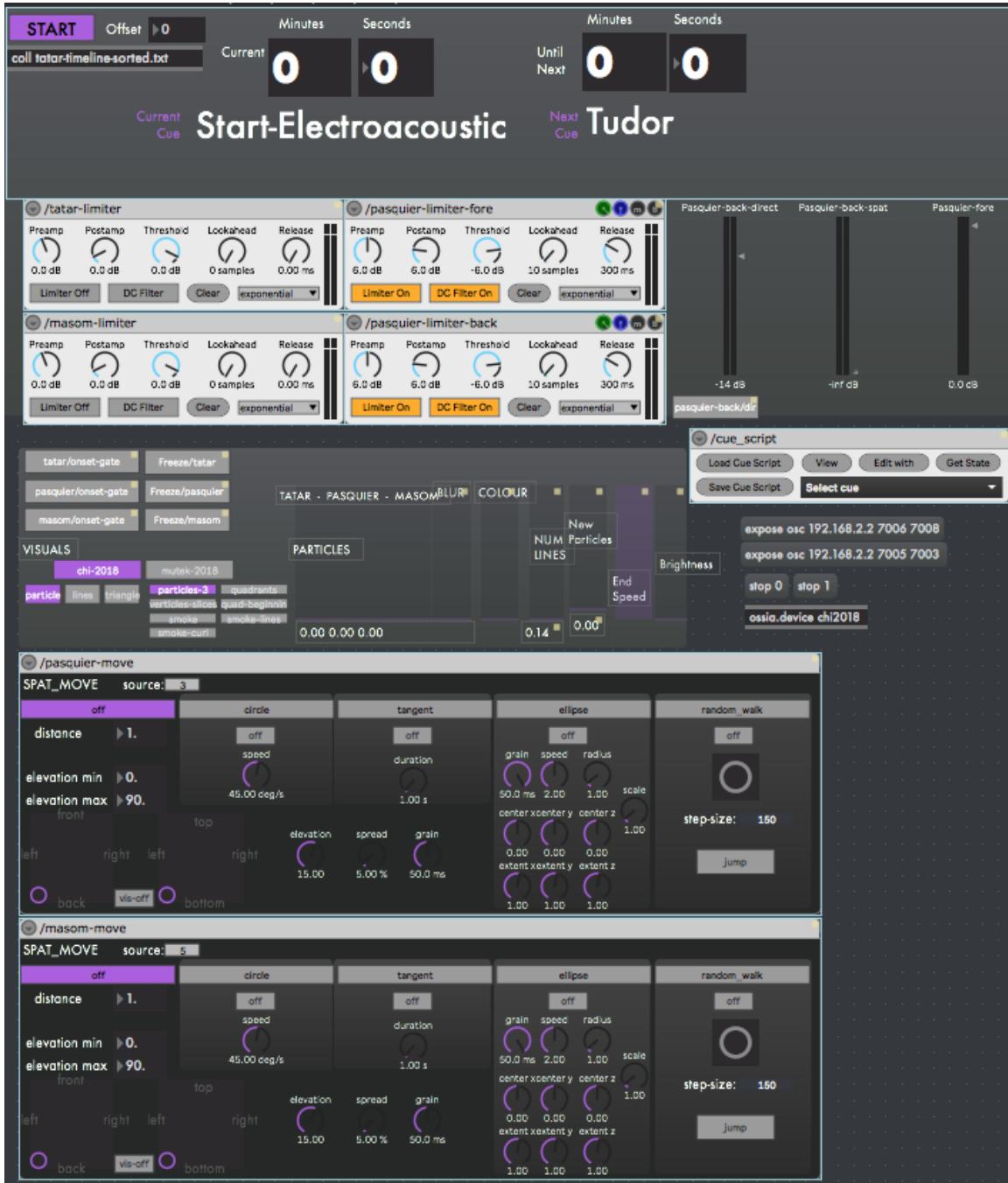
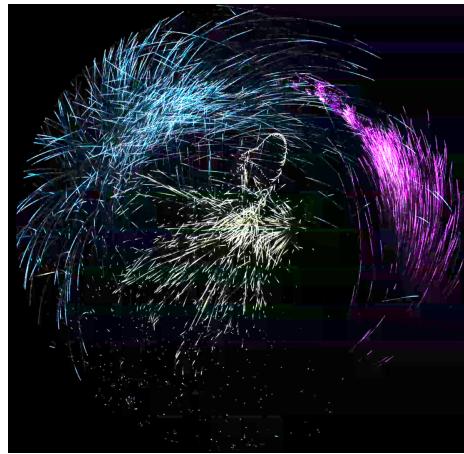
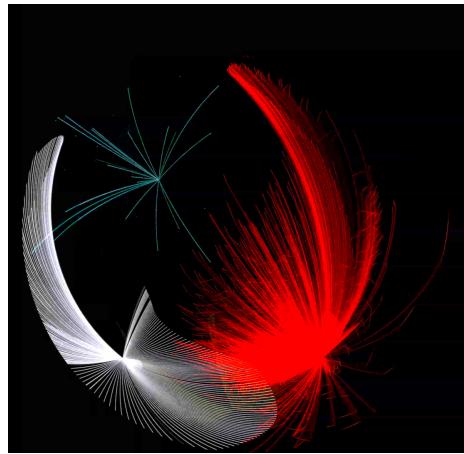


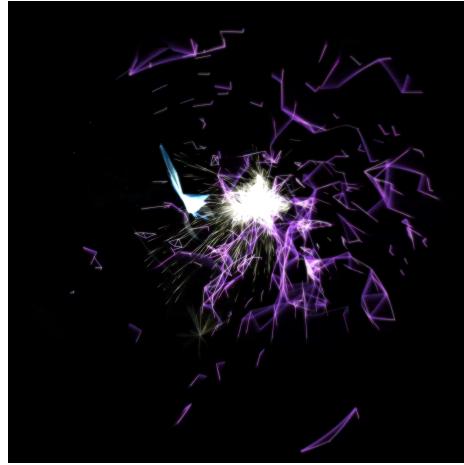
Figure 5.1: The UI of Revive's performance setup



(a) Rendering type 1 - particles as sprites



(b) Rendering type 2 - particles for drawing lines



(c) Rendering type 3 - particles for drawing triangular shapes

Figure 5.2: Three snapshots of dome views illustrate three types of particle engine renderings. In each figure, three distinct colors corresponds to three visual agents that react to three sonic performers.

behaviours, three visual agents emphasize the actions of audio agents and make it easier for the audience to comprehend the connection between sonic gestures and the sonic performers' actions.

The visual agent architecture consists of a machine listening module for audio feature extraction and a particle engine to generate visuals. The location of particle engines' rendered outputs are synchronized with the spatial location of sonic performers in the 3D audio setup. As the sonic performers move in the 3D audio, the visuals follow the locations of sonic performers. During the Revive performances, three types of particle renderings provide variety in the visual content: 1- particles as sprites, 2- particles for drawing lines, 3- particles for drawing triangular shapes (Figure 5.2). The audio-reactive behaviours of three particle engines apply mappings of sonic performers' audio features to the input parameters of particle engines.

The machine listening modules of visual agents implement an onset detection with onset loudness detection and calculate continuous total loudness and specific loudness [15]. The onset detection uses the magnitude spectrum and if the summation of magnitude powers of all bands pass a user-set threshold, an onset is detected. Specific loudness is the loudness calculated for the Bark bands that is a 24-band spectrum that approximates critical bands of human hearing. We also apply a post-processing on the specific loudness to calculate a 3-band loudness spectrum of bass, middle, and high bands. The first two Bark bands are combined for the bass spectrum, the last eight bands are joined for the high spectrum, and the remaining fourteen bands constitute the middle band. The audio features are calculated for all sonic performers separately.

The audio features of a sonic performer are mapped to the parameters of corresponding particle engine. The total loudness is mapped to the overall particle speed and the brightness of particles. Two color schemes are applied: 1- static per performer, 2- the loudness of the mid-band controls the amount of green in red-green-blue color representation. Additionally, if the total loudness passes a certain threshold (above -3 dB for example), the type of noise that scatters the particles changes. This increases the overall movement of particles when the loudness moves closer to the maximum. Regarding rendering type 2 and 3 (Figure 5.2b and 5.2c), the particles have a life-span which is also a gaussian distribution. A static noise generates the birth locations of particles and the attraction points that the particles move towards within their life-time. The scale of birth locations is linearly mapped to the loudness of the bass spectrum. When there is more variance in the bass spectrum, the distances between the particle birth locations and the attraction points increase, which results in an increase in the fast movement behaviors of particles. The particles are re-initiated when the loudness of an onset passes a -6 dB threshold. As we mention in Section 5.3, the second author also plays a background channel that directly outputs to all speakers. Lastly, the total loudness of this channel controls the amount of post-processing effects such as blur and feedback applied to all visuals.

5.5 Sonic Strategies in Revive

The first and the second author join MASOM in the sonic performance of Revive. These three performing agents apply different techniques and approaches, which come together through pre-

defined roles within the structured improvisation. *Revive*'s aesthetics allow performers to improvise within these roles, and this introduces live sonic gestures back to the performance. In the following sections, we delve into the techniques of three sonic performers.

5.5.1 Musical Artificial Intelligence, MASOM-FO

Varèse defines music as “nothing but organized sounds” [21]. Inspired by this idea, MASOM’s system design implements a neural networks algorithm combined with statistical sequence modelling algorithms in Max. The neural networks algorithm organizes the sound memory of the agent whereas the statistical sequence modelling algorithms handle the temporal musical structure modelling and user interaction. MASOM’s unsupervised learning requires a set of audio recordings. The agent implements a Music Emotion Recognition algorithm in the machine listening module.

MASOM applies offline learning and online generation. The offline learning starts with segmentation of individual sounds in the recording (Figure 5.3a and 5.3b). The agent recognizes the audio segments between onsets as audio samples. MASOM use spectral magnitude based onset detection where an onset is detected when the summation of spectral magnitudes passes a user defined threshold. The implementation of segmentation and audio feature extraction uses IRCAM’s MuBu Max Package⁶ [16] and PiPo externals⁷. Following the segmentation, the training procedure labels audio samples with a 35-dimensional audio feature vector including:

- Timbre features: Perceptual Spectral Decrease and 13 Mel-frequency Coefficients (MFCCs)
- Fundamental Frequency
- Loudness
- Duration of audio sample
- Music Emotion Recognition (MER) features

Regarding timbre features, loudness, and fundamental frequency; we first calculate the features using a window size of 1024 samples and hop size of 256 samples, then we calculate the mean and standard deviation of these features per audio sample. The statistics of loudness, and 13 MFCCs, perceptual spectral decrease, and YIN-based fundamental frequency estimation⁸ adds up to 32 features $((1 + 13 + 1 + 1) * 2 = 32)$. In addition, we add the duration of audio samples, and two MER features and the total number of audio features constitutes a 35-dimensional label vector.

⁶<https://forumnet.ircam.fr/product/mubu-en/>

⁷<http://ismm.ircam.fr/pipo/>

⁸Please refer to [15] for the details of audio feature calculations and [4] for the YIN algorithm.

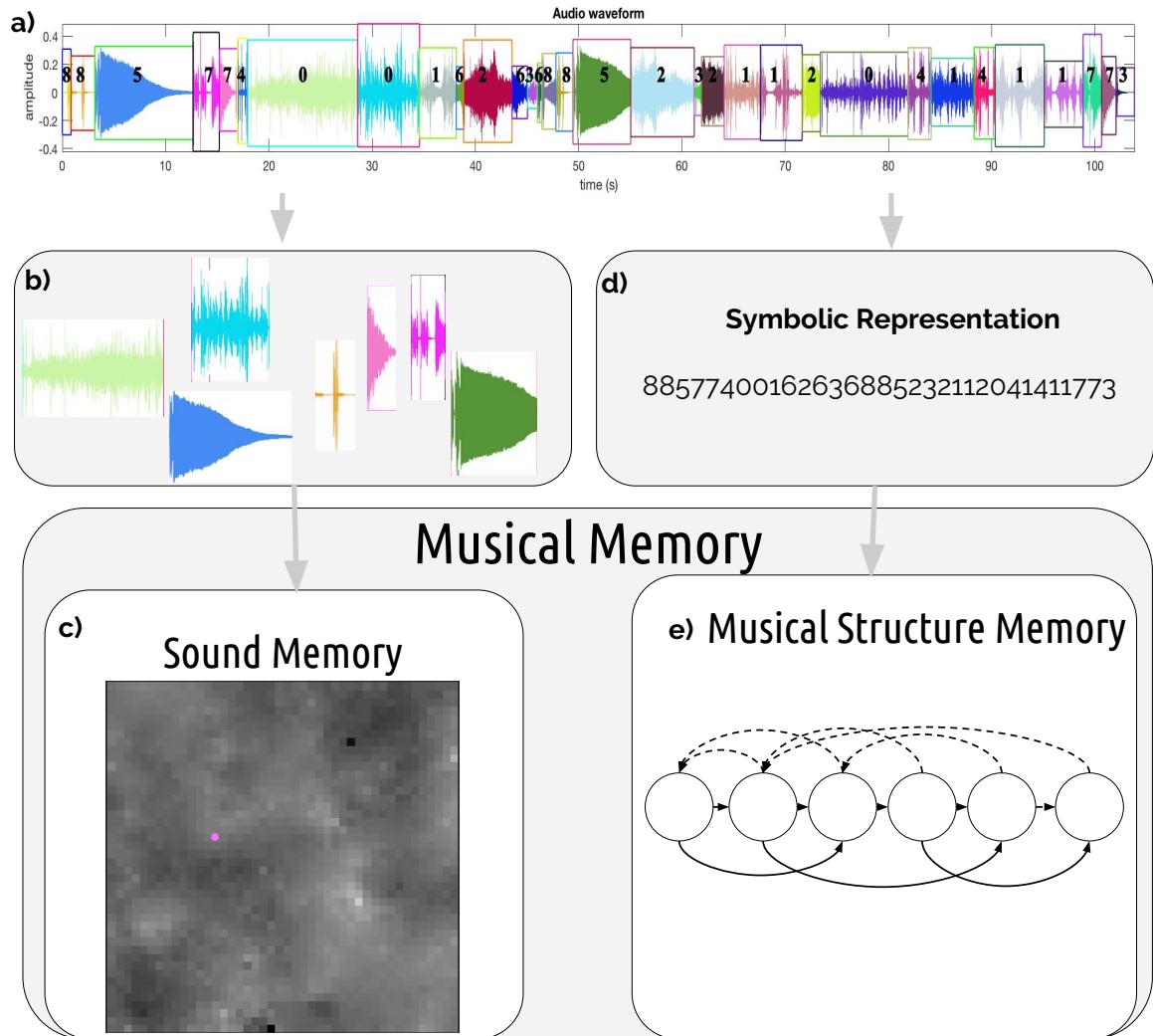


Figure 5.3: The offline learning in MASOM: a) Segmentation b) Labelling the audio samples c) Sound memory where squares stand for SOM nodes that are a clusters of audio samples d) Creating a symbolic representation of the original song using the clusters indexes of audio samples e) statistical sequence model to learn temporal transitions.

The particular MER model that we use in the system design of MASOM is a two-dimensional, continuous multivariate linear regression model using the following equations:

$$\begin{aligned}
 Pleasantness = & - 0.169 + & (5.1a) \\
 & - 0.061 * LoudnessMean \\
 & + 0.588 * SpectralFlatness1Mean \\
 & + 0.302 * MFCC1STD \\
 & + 0.361 * MFCC5STD \\
 & - 0.229 * PerceptualSpect.DecreaseSTD
 \end{aligned}$$

$$\begin{aligned}
 Eventfulness = & - 1.551 & (5.1b) \\
 & + 0.060 * LoudnessMean \\
 & + 0.087 * LoudnessSTD \\
 & + 1.905 * PerceptualTristimulus2STD \\
 & + 0.698 * PerceptualTristimulus3Mean \\
 & + 0.560 * MFCC3STD \\
 & - 0.421 * MFCC5STD \\
 & + 1.164 * MFCC11STD
 \end{aligned}$$

We generated this MER machine learning model using the Emo-soundscapes dataset that contains 600 curated soundscape recordings [7]. In the audio domain, the terms valence and arousal is exchanged with eventfulness and pleasantness, respectively [6]. This is mainly because of the contradiction that a sound does not feel an emotion, but stimulate an emotion. Hence, we can't label a sound with emotion categories of human cognition. For example, we can't talk about a happy sound (excluding anthropomorphism), but we can say that some sounds initiate happiness feelings in humans.

Following the segmentation and labelling audio samples, the agent trains a Self-Organizing Map to create a latent space of sonic possibilities (Figure 5.3c). In this latent space, similar sounds locate close to each other. Self-Organizing Maps are fully connected artificial neural networks with unsupervised learning[9, 10]. SOM is used for visualisation, representation, and clustering of high-dimensional input data with a 2D topology of square, rectangular, toroid, and arbitrary shapes (such as Mnemonic SOMs). SOMs consist of a number of nodes that position themselves during training to represent the topology of the input data. The nodes are vectors with the same number of dimensions with the input data. Hence, SOM creates a symbolic latent space that represent the topology of the training data.

MASOM incorporates the SOM implementation included in *ml.star* Max Package [17]. Regarding the SOM training ⁹, we first normalize the input data using the equation 5.2:

$$I_{norm}[i] = \frac{I[i]}{M[i]} * STD[i] \quad (5.2)$$

and $i \in [1 : N]$, where the N is the total number of dimensions of the input vector, I is the calculated audio feature vector, M is a vector that gives the average of each audio feature, STD is a vector of standard deviation of each audio feature, and I_{norm} is the normalized feature vector. M and STD are calculated for a given corpus.

Then, we apply a weight vector on the normalized input data of SOM. In the weight vector W , MFCC features are multiplied by 1/13 so that combined MFCC distances affect the SOM training as one timbre feature. The rest of the features are kept at the original value.

$$a = \text{int}(\sqrt{\text{the number of audio samples in the memory}/6}) \quad (5.3)$$

After the the pre-processing, we train an SOM map with square topology, $a * a$ where a is found by using the equation 5.3 that aims for 6 samples per cluster. We found that this approach consistently gave a low number of SOM nodes where no audio sample was clustered. The total number of epochs in the SOM training is 1000. The learning rate and the neighbourhood drops from 0.25 to 0.01 and $a/4$ to 0, respectively.

Clustering follows the training of SOM sound memory. We calculate Euclidean distance between SOM node vectors and thumbnail vectors of audio samples in the agent's memory. The audio samples are labelled with the SOM node that gives the lowest Euclidean distance. The numbers within the segment squares in Figure 5.3a are the closest SOM node indexes. Following this labelling, we generate a symbolic representation of the original song using the sequence of SOM node indexes (Figure 5.3d). This sequence is later used to train the statistical sequence modelling algorithm.

The agent can generate compositions on the fly and change musical structure models through user interaction. A statistical sequence modelling algorithm learns and generates SOM node index sequences (Figure 5.3d). These nodes are clusters of audio samples, and the agent chooses a sample clustered by an SOM node randomly. MASOM's training aims for 6 audio samples per node, and the sonic variation within a node is constrained because of the unsupervised learning. Hence, the random sample selection within the SOM node audio cluster implements a type of constrained sonic variation in the agent's output.

MASOM's architecture was initially using Variable Markov Models (VMM) Prediction by Partial Matching C (PPM-C) variant [18]. MASOM VMM PPM-C variant was trained on small and medium size audio corpora ranging from an album to approximately 1GB of audio recordings (lossless stereo wave files). Although the interactive nature of MASOM VMM PPM-C variant was satis-

⁹The details of SOM training procedures is available at [10, 18].

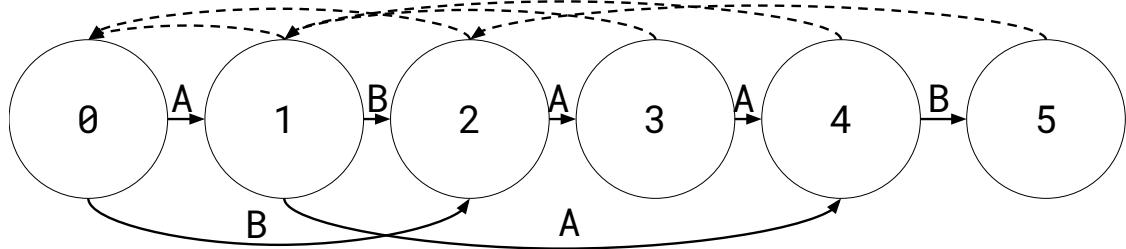


Figure 5.4: Factor Oracle generated using the sequence *ABAAB*.

fying enough for public concerts with small and medium size corpora, we realized that this variant lost timbre consistency with big size corpora ranging from 1GB to 100GB of audio recordings. Hence, we switched MASOM’s statistical sequence model to a generative Factor Oracle algorithm to ensure timbre consistency and we refer to this variant as MASOM-FO.

Factor Oracle, initially proposed as a compression algorithm, is a statistical sequence modelling algorithm as well as a finite state automata [2]. FO models repeating patterns in a sequence, that is, the *factors* of a sequence. FO has three types of links: internal links (forward links between successive states), external links (forward links that jumps to a future state), and suffix links (backward links, dashed lines in Figure 5.4). Suffix links marks the longest repeating factor in previous states. FO allows incremental learning, and learning is linear in time and space [11]. Several musical agents previously implemented FOs in the architecture [19, p. 26-29].

The training of FO allows a single sequence for training. In MASOM’s case, the agent’s corpora include several audio recordings and the symbolic SOM node representations of these recordings¹⁰. This constraint emerges two options for FO training: 1- concatenating several (or all) symbolic representations of audio recordings to one sequence, 2- using the symbolic representation of only one recording. The first option risks timbre consistency as various audio recordings cover a wide range of timbre possibilities, even within the same musical style. Hence, we apply option two in Revive project to ensure timbre consistency.

MASOM-FO incorporates Wilson’s [22] Max external implementation which is sufficiently fast for real-time training. In Revive’s structured improvisation, MASOM’s role in the performance is constrained by the FO training. In each section, we train the agent’s FO from scratch using the symbolic representation (the sequence of sound cluster indexes) of one audio recording. This results in a high-level user interaction where the user defines the musical constraints and roles of the agent by forcing a subset of sound clusters, and the temporal patterns and structures of an audio recording. The subset of sound clusters may contain samples of other recordings due to the SOM training, and this introduces variety in the agents audio output.

FO initiate SOM node (cluster) index generation with sound selection using two approaches: 1- playing one sample after another, 2- user-defined time intervals. The first approach outputs one

¹⁰The details of FO training is available in [22].

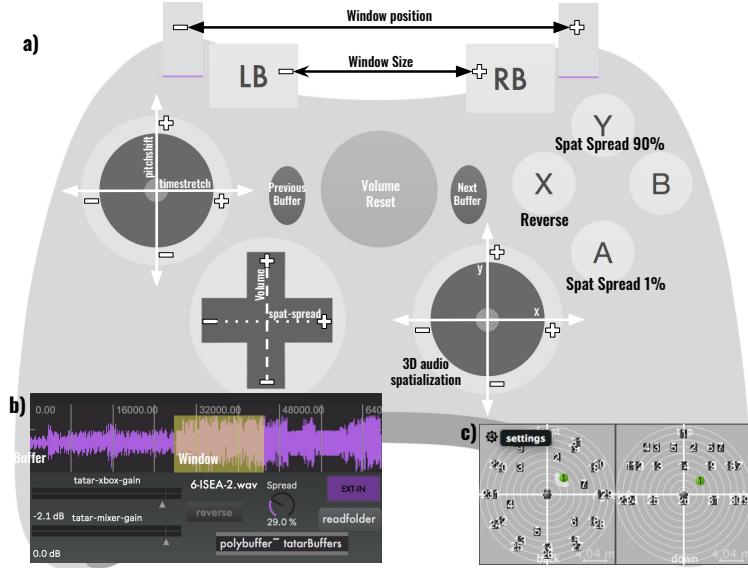


Figure 5.5: The framework for using fixed-media compositions as content for the live audio-visuals with 3D audio: a) the game controller interface and the mappings b) wavetable synthesis c) 3D audio visualization with the SAT dome setup.

layer of sound events where each sample is concatenated one after another. This approach generates a monophonic output as the FO waits for the previous sample to finish before initiating the next one. The second approach creates a multi-layered output where the user can change the audio event density by manipulating the time intervals between sample initiations. The agents uses FO to output a sound cluster index in user-defined intervals. Using any of these two approaches, FO outputs a sound cluster index. MASOM chooses a sample within that cluster randomly. The random picking of samples is a constrained selection because of the SOM clustering, and it aims for the generation of constrained variation in the agent's output. In Revive, these two generative approaches are initiated by performance cues. We carry out the audio event density manipulation in the second approach by applying linear ramps on the time interval parameter.

5.5.2 Distruption of fixed-media

Acousmatic music facilitates electronic means to create or process sounds to produce musical compositions. The public presentations of acousmatic music happens as playback of the compositions without any interaction, or live mixing of prepared tracks using volume adjustments, fade ins and outs, equalizers, and spatialization [13]. The second author puts a rich MAX-based sampler player, Kenaxis¹¹ into practice with samples of analog machines such as EMS Synthi AKS, Korg Mono/Poly, Serge and Eurorack analog modular synthesizers, as well as field recordings of forests, waterfalls, and thunderstorms. Samplers and granular synthesis engines are controlled live with a

¹¹<https://www.kenaxis.com/products/kenaxis-2/>

BCF-2000 MIDI controller. The setup allows a sonic palette of background layers made of drony textures, as well as foreground gestures including more melodic motifs with live effects.

The reflections of the third wave of HCI studies have initiated the introduction of embodied interaction for live music performances [5] and proposed a combined understanding of sonic and bodily interaction [8]. In line with the live performance trends in computer music, the first author's live sonic performance transforms his fixed-media compositions to sonic material for improvisation combined with live generation of sonic gestures. In this self-disruption process, the first author metamorphoses his previous fixed media pieces to a sonic vocabulary for live improvisation of experimental electronic music. To do so, the first author combines wavetable synthesis with a game controller interface to improvise experimental electronic music on a 3D speaker configuration (Figure 5.5).

Wavetable synthesis uses varying playback speeds to manipulate an audio buffer to synthesize sounds. Varying playback speed generates pitch-shifting and time-stretching audio manipulations, and it is possible decorrelate these two manipulations [1]. The wavetable synthesis utilize an audio buffer and allows sudden changes of the audio in the buffer. The first author's performance practice in Revive applies one-minute long excerpts from his previous fixed media compositions (Figure 5.5b). The audio buffer of wavetable synthesis is a certain portion (window) of these one minute excerpts. This approach provides a sonic diversity using varying window sizes and positions using a game controller as the interface.

The first author utilizes an XBox controller that includes two joysticks buttons underneath, one d-pad, 9 additional buttons (x, y, a, b , left-button, and right-button), and two triggers that acts as sliders, illustrated in Figure 5.5a. The *joystick-2* handles the interaction with the spatialization and the rest manipulates the wavetable synthesis. Although the *joystick-2* outputs a 2-dimensional spatial position, we project 2D positions on a 3D spherical surface to create 3D spatialization trajectories in azimuth-elevation-distance format using the equations,

$$r_{2D} = \sqrt{(position_x_{2D})^2 + (position_y_{2D})^2} \quad (5.4a)$$

$$azimuth_{3D} = \arccos\left(\frac{position_x_{2D}}{r_{2D}}\right) \quad (5.4b)$$

$$elevation_{3D} = \frac{\pi}{2} * (1 - r_{2D}) \quad (5.4c)$$

where r_{2D} is the distance from the center in 2D, $azimuth_{3D}$ and $elevation_{3D}$ ranges are $[-\pi, \pi]$ and $[0, \frac{\pi}{2}]$; respectively. The distance is static, and this ensure 3D spatialization while preserving the original loudness of the audio source.

The y-axis of *joystick-1* controls the amount of pitch-shifting, and the x-axis controls the multiplier of time-stretching. When *joystick-1* is in the resting position, the wavetable synthesis plays the original content in the audio buffer. The button on the *joystick-1* plays the sound; hence, there is no sound coming out when the user moves the hands away from the interface. The button x reverses the audio buffer. The button b allows a toggle for continuous audio play for occasion where

a sustained continuous playback is needed, such as textural sounds that functions in the sonic background. Hence, the performer can explore a continuum of sonic choices from gestural foreground actions to textural background material. The up and down buttons on the *d-pad* adjust the output volume, and left and right buttons change the source spread in the 3D spatialization. The spread controls the total area of audio source diffusion. The buttons *a* and *y* set the spread to 1% and 90% in 2-seconds, respectively. The performer can cycle through several audio buffers using *start* and *back* button. Lastly, the *home* button resets the output volume to 0 dB in 3-seconds.

5.5.3 3D audio spatialization techniques in Revive

Sonic performers in Revive are spatialized in 3D speaker setups using IRCAM-SPAT Max library¹² [3]. This library is fast and flexible to change 3D audio speaker setups and test spatialization methods in soundchecks and rehearsals where the time is limited. In Revive, we use 3D vector-based amplitude panning (vbap3D). The audio source positions are processed using the azimuth-elevation-distance format. The distance is static so that performers change the loudness of their output as they prefer. Throughout this paper, the elevation range is $[0, \frac{\pi}{2}]$ that addresses the setup of the SAT dome. The elevation range is an input variable for all spatialization modules for different 3D speaker setups.

The first author controls the spatial location using the joystick on the game controller. The output of second author and MASOM are positioned using three different generative trajectories: circular, tangential, and random-walking. The movement speed is a variable in all three cases, and it is changed with the cue system of Revive. The first method cycles the azimuth using a speed variable (degree/s) and the elevation can be fixed or adjusted by the cue system. The second approach, tangential trajectory generation chooses a random location on the opposite half of the sphere in relative to the current source position, and the source moves to this location in a duration that is set by the cue system. The azimuth and elevation of the new location is generated using the following formula:

$$\begin{aligned} azimuth_{new} = & \\ random(azimuth_{current} + \frac{\pi}{2}, azimuth_{current} + \frac{3\pi}{2}) \end{aligned} \tag{5.5a}$$

$$\begin{aligned} elevation_{new} = & \\ (elevation_{current} + random(0, \frac{\pi}{4})) \bmod \frac{\pi}{2} \end{aligned} \tag{5.5b}$$

where the function *random*(*a*, *b*) generates pseudo-random values in the range of $[a, b]$. The third approach, random-walking first generates a step size within a user-set range using the pseudo-

¹²<https://forumnet.ircam.fr/product/spat-en/>

random function. The generated step size is added to the current location of the source; and thus, the source jumps to a new location. The steps in random-walking can be triggered by two ways: using fixed durations or using a magnitude spectrum based onset-detection. In case of MASOM, we apply an additional approach where the random-walking is triggered with every sample initiation in MASOM. The trajectory generation selections and their input variables are controlled by the cue system during the performance.

5.6 Conclusion

We introduced the live audio-visual performance project *Revive*. This live performance benefits from an automatized parameter management using Jamoma’s cue system. The compositional approaches of *Revive* allows performers to improvise within predefined roles. The medium of this audio-visual performance project is a dome projection with 3D audio setup where the audio and the visuals are synchronized in position.

Revive fuses a musical AI architecture into structured improvisation for audio-visuals. The musical AI in Revive, MASOM-FO employs Factor Oracle algorithm for temporal content generation and user interaction. Thomas et al. [20] compare Factor Oracle, Fixed-Length Markov Model, and MusiCOG on melody generation and show that these sequence modelling algorithms introduces particular biases to the sequence generation. In our future work, we plan to compare various statistical sequence modelling algorithms in terms of plagiarism, that is how much the model copies the patterns in the training dataset.

During the development of *Revive*, the task with the highest time-complexity was the setup and exploration of the mapping between audio features and input parameters of generative visuals. Although several mapping tools have been developed previously, these tools are not necessarily made for exploration and comparison of several mapping possibilities. To address this issue, several researchers around the globe planing to collaborate with the OSSIA¹³ initiative. OSSIA, stands for Open Software System for Interactive Applications, is an OSC-query based open-source framework for time-scripting and mapping for interactive scenarios. The framework is currently in its alpha stage, and we hope that the project will move further in future.

Bibliography

- [1] zynaptiq: ZTX Features And Specifications. URL <https://www.zynaptiq.com/ztx/ztx-features-and-specifications/>.
- [2] Cyril Allauzen, Maxime Crochemore, and Mathieu Raffinot. Factor oracle: A new structure for pattern matching. In *International Conference on Current Trends in Theory and Practice*

¹³<https://ossia.io/>

- of Computer Science*, pages 295–310. Springer, 1999. URL http://link.springer.com/chapter/10.1007/3-540-47849-3_18.
- [3] Thibaut Carpentier, Markus Noisternig, and Olivier Warusfel. Twenty Years of Ircam Spat: Looking Back, Looking Forward. In *Proceedings of the 41st International Computer Music Conference (ICMC 2015)*, pages 270–277, Dentan, Texas, USA, 2015.
 - [4] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917, 2002. ISSN 00014966. doi: 10.1121/1.1458024.
 - [5] Paul Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. The MIT Press, Cambridge, Mass., 1 edition edition, August 2004. ISBN 978-0-262-54178-7.
 - [6] Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society*, 64(9):646–653, September 2016. ISSN 15494950. doi: 10.17743/jaes.2016.0044.
 - [7] Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Emo-soundscapes: A dataset for soundscape emotion recognition. In *2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 196–201, San Antonio, TX, October 2017. IEEE. ISBN 978-1-5386-0563-9. doi: 10.1109/ACII.2017.8273600.
 - [8] Alexander Refsum Jensenius and Michael J. Lyons, editors. *A NIME Reader: Fifteen Years of New Interfaces for Musical Expression*. Current Research in Systematic Musicology. Springer International Publishing, 2017. ISBN 978-3-319-47213-3.
 - [9] Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982.
 - [10] Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1–3):1–6, November 1998. ISSN 0925-2312. doi: 10.1016/S0925-2312(98)00030-7.
 - [11] Arnaud Lefebvre and Thierry Lecroq. A Heuristic For Computing Repeats With A Factor Oracle: Application To Biological Sequences. *International Journal of Computer Mathematics*, 79(12):1303–1315, January 2002. ISSN 0020-7160, 1029-0265. doi: 10.1080/00207160214653.
 - [12] Trond Lossius, T. de la Hogue, Pascal Baltazar, Timothy Place, N. Wolek, and Julien Rabin. Model-View-Controller separation in Max using Jamoma. In *Proceedings of the joint 40th International Computer Music Conference & 11th Sound and Music Computing Conference*,, page 8, Athens, Greece, 2014.
 - [13] Eduardo Reck Miranda and Marcelo Wanderley. Toward Intelligent Musical Instruments. In *New Digital Musical Instruments: Control And Interaction Beyond the Keyboard*, pages 219–

255. A-R Editions, Inc., Middleton, Wis, 1st edition edition, July 2006. ISBN 978-0-89579-585-4.

- [14] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An Introduction to Musical Metacreation. *Computers in Entertainment*, 14(2):1–14, January 2017. ISSN 15443574. doi: 10.1145/2930672.
- [15] G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, 2004.
- [16] Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Riccardo Borghesi. MuBu and friends-assembling tools for content based real-time interactive audio processing in Max/MSP. In *Proceedings of International Computer Music Conference (ICMC)*, 2009.
- [17] Benjamin D. Smith and Guy E. Garnett. Unsupervised Play: Machine Learning Toolkit for Max. In *the Proceedings of International Conference on New Interfaces for Musical Expression 2012*, 2012.
- [18] Kivanç Tatar and Philippe Pasquier. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, June 2017. ISBN 978-1-77287-019-0.
- [19] Kivanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):1–50, September 2018. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1511736.
- [20] Nicolas Gonzalez Thomas, Philippe Pasquier, Arne Eigenfeldt, and James B. Maxwell. A Methodology for the Comparison of Melodic Generation Models Using Meta-Melo. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 561–566, Brazil, 2013. ISBN 978-0-615-90065-0.
- [21] Edgard Varèse and Chou Wen-chung. The liberation of Sound. *Perspectives of New Music*, 5 (1):11–19, 1966.
- [22] Adam James Wilson. factorOracle: an Extensible Max External for Investigating Applications of the Factor Oracle Automaton in Real-Time Music Improvisation. In *Proceedings of the International Workshop on Musical Metacreation 2016*, page 5, Paris, France, 2016.

Chapter 6

Respire: A Virtual Reality Art Piece with a Musical Agent guided by Respiratory Interaction

KIVANÇ TATAR

MIRJANA PRPA

PHILIPPE PASQUIER

AS ACCEPTED TO THE LEONARDO MUSIC JOURNAL ON JANUARY 16TH, 2019, IN
PRESS

Abstract

Respire is an immersive art piece that brings together three components: an immersive virtual reality (VR) environment, embodied interaction (via a breathing sensor), and a musical agent system to generate unique experiences of augmented breathing. The breathing sensor controls the user's vertical elevation of the point of view under and over the virtual ocean. The frequency and patterns of breathing data guide the arousal of the musical agent, and the waviness of a virtual ocean in the environment. Respire proposes an intimate exploration of breathing through an intelligent mapping of breathing data to the parameters of visual and sonic environments.

Keywords: Musical agents; Multi-Agent Systems; Artificial Intelligence; Musical Metacreation; Computational Creativity; Virtual Reality; Interactive Arts; Contemporary Arts



Figure 6.1: The virtual environment visuals in *Respire*

6.1 Immersive Environment of Respire

Respire (2018), built on our previous work *Pulse.Breath.Water* (2016), immerses the user in a virtual environment depicting an ocean, and the user traverse the environment using their breathing (Figure 6.1). The environment, made with Unity and presented with HTC Vive, evokes a dark, gloomy atmosphere with the elements like fog and waves that wrap around the user [4]. Ambiguity of the visuals and the lack of focal objects stimulate the user to engage in the process of sense-making of the scene. Ambiguity in design of interactive artifacts engages users to project their own values and experiences in the process of meaning-making of the visual stimulus [8]. *Respire* also exercises a minimalist color scheme with grayscale and a subtle blue to elicit the effect of beholder's share. Kandel [9] proposes beholder's share as a process of the user's projection of previous experiences in sense-making of an ambiguous scene. By immersing the user in an ambiguous environment, *Respire* aims to give the user a canvas to "paint" their own experiences. The immersion avoids imposing an explicit narrative and the constraints of a story. Hence, the environment empowers the user to curate their experiences.

6.2 Body and breath in immersive virtual environments

Cartesian dualism approaches a human being as a "thinking thing" that is divorced from bodily experience. This notion sparks new discourses on what does it mean "to be" in the environment.

The examples of embodied interaction challenge the Cartesian separation between a subject and object, and this separation translates into artistic practices as a separation between the artwork and the audience. Embodied interaction [6] emphasizes the value of engaging our bodies in interaction and transcends Cartesian dualism by focusing on the body along with the mind as a united medium to experience the environment.

What could that link between this united medium and the environment be? Artists explored breathing as a connector between the body and virtual environments generated using computational means. By positioning the body in the center of the artwork, and employing breath in embodied interaction paradigm, artists succeeded to create that tangible yet invisible link. For instance, Sonia Cillari's "As an artist, I need to rest" (2009) explores how a body can be source of artificial-life (a-life) through breathing [10]. Cillari employs breathing data to generate a virtual environment of feathers, and maps the breathing patterns to the movement of the feathers in the virtual environment.

Likewise, Char Davies' pioneering piece, "Osmose" (1995) is an immersive virtual environment presented on a head-mounted display (HMD) [5]. The user navigates the movement in the virtual environment with the breathing and body balance. The breathing controls the elevation whereas the body balance changes the horizontal 2D direction. This mapping resembles the experience of diving, and likewise, we are inspired by diving phenomena in the breathing interaction of Respire. Davies juxtaposes two ideas: one of immateriality of computer generated worlds and body-felt phenomena elicited by those environments. The sense of virtual presence afforded by VR medium opened a dynamic space for artistic explorations as demonstrated in Davies' piece [5]. Respire continues that space for artistic exploration by introducing agent architecture to react to breathing.

6.3 Artificial Intelligence, Multi-Agent Systems, and Musical Agents

All humans breathe, consciously or unconsciously. As a substantial element of being alive, breathing continues even when we do not attend to the act of breathing. Our control of breathing shifts depending on our attention. Inspired by this mechanism, Respire intends to use advance computational tools for an intelligent mapping from breathing to movement, sound, and visuals; so as to elicit attention and mindfulness. Artificial Intelligence (AI) and Multi-Agent Systems (MAS) provide such tools for computational creative applications [12].

The agent paradigm appears in many disciplines such as Social Sciences, Philosophy, Cognitive Science, and Computer Sciences. In Computer Sciences, an agent is an autonomous system that initiates actions to respond to its environment in timely fashion [18]. Multi-agent Systems (MAS) studies the agent architectures for computational applications. Musical agents are artificial agents that automatize musical creative tasks [17]. Respire's architecture implements this intelligent mapping using a musical agent system.



Figure 6.2: The system architecture of *Respire*

6.4 Affect Recognition

Affect Recognition focuses on designing computational models that can estimate the affective state of a content. For example, the content can be an image, a video, a human body posture, a sound, or a music piece. Mainly, two types of affective models appear in affective computing: discrete or continuous [11]. *Respire* implements a two-dimensional continuous affect model that is presented by Tatar and Pasquier [16]. Dimensional affect estimation models generate a bounded, continuous output to which we apply signal processing, mapping and generative algorithms.

Respire computationally generates the visual and sonic environments using two separate frameworks: a VR system and a musical agent generating the sonic environment. These frameworks utilize affective dimensions in the system architectures. The generative content and reactive behaviors of these systems use affective dimensions as a high-level cross-medium paradigm for intelligent mapping. This enables a human-readable parametrization of two systems generating visuals and audio separately.

6.5 System Details

Respire aims to bridge the virtual presence and innate experiences of the user. The artwork builds upon breath-based embodied interaction, and utilizes a breath controller (Thought Technology Pro-Comp2 with the respiration harness) for the user's vertical position in the scene, simultaneously allowing for exploration of the virtual environment and breathing patterns (Figure 6.2). This mapping of breathing to the vertical elevation resembles diving to guide the interaction so that the audience does not need to learn something new to participate in the artwork. We previously compared this mapping with other options, and found the current mapping to be intuitive from the user's perspective [13]. Also, rapid breathing creates more eventful sonic environment and reflects in the ocean surface filled with waves. Less eventful breathing (slow paced breathing) calms the ocean surface and generates a calmer sonic environment. The pleasantness dimension calculated by the musi-

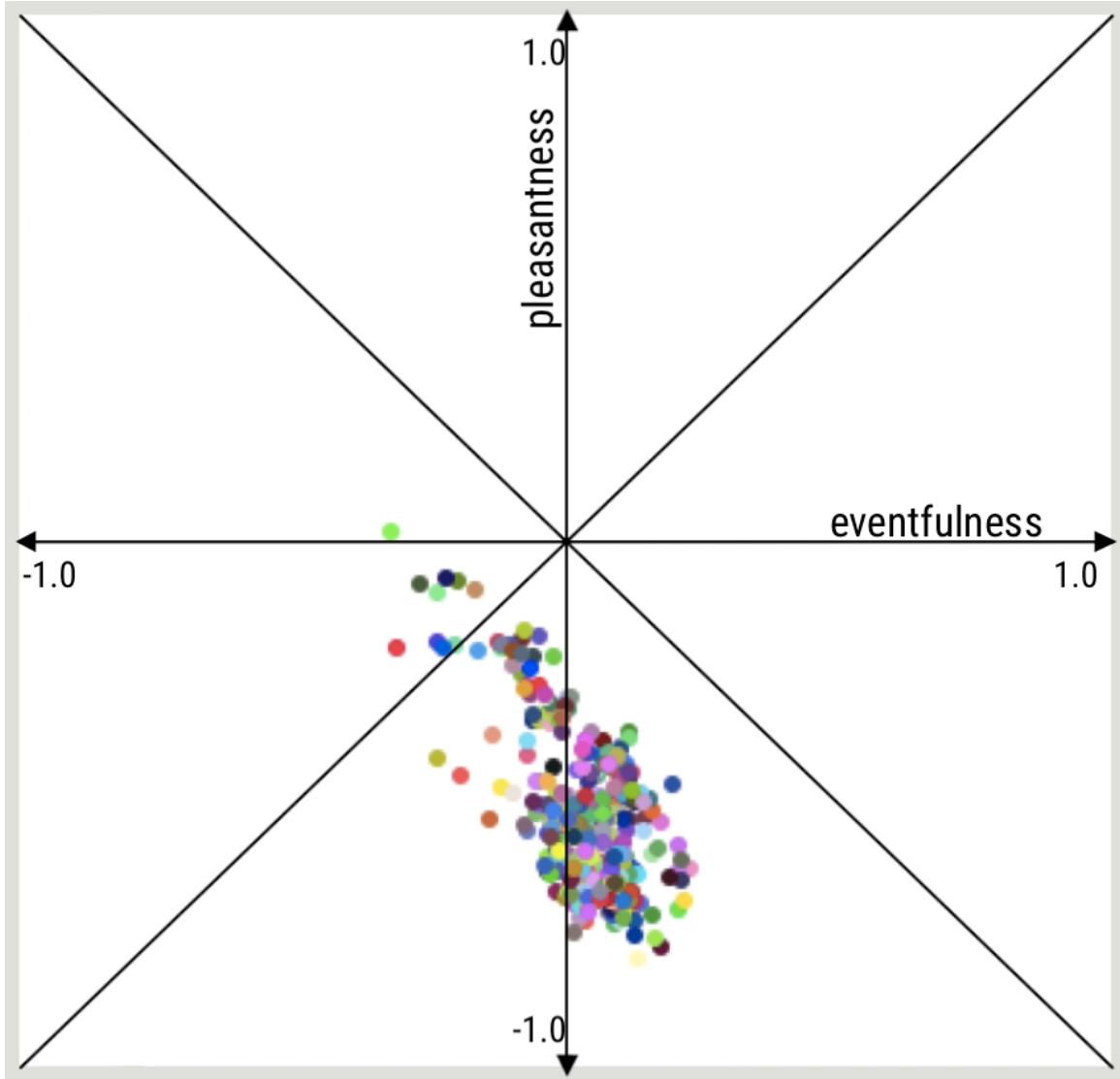


Figure 6.3: The system architecture of *Respire*

cal agent (see Figure 6.2), controls the color of the sky in the environment, turning the sky in the range between black (low pleasantness values) to white (high pleasantness values).

The musical agent is developed in Cycling74's Max and the communication between Max and Unity utilizes UDP-based OSC. The breathing sensor data is passed to Max using M+M middleware [3]. In the following, we delve into the agent architecture that consists of five main modules: memory, perception, goal, action, and post-processing.

6.5.1 Sound Memory

The agent's memory consists of audio samples and symbolic data of pleasantness and eventfulness of the samples (Figure 6.3). This type of corpus known as a hybrid corpus in musical agents [17]. In line with Respire's aesthetic choice of ambiguity, we aimed for a curation of abstract and ambi-

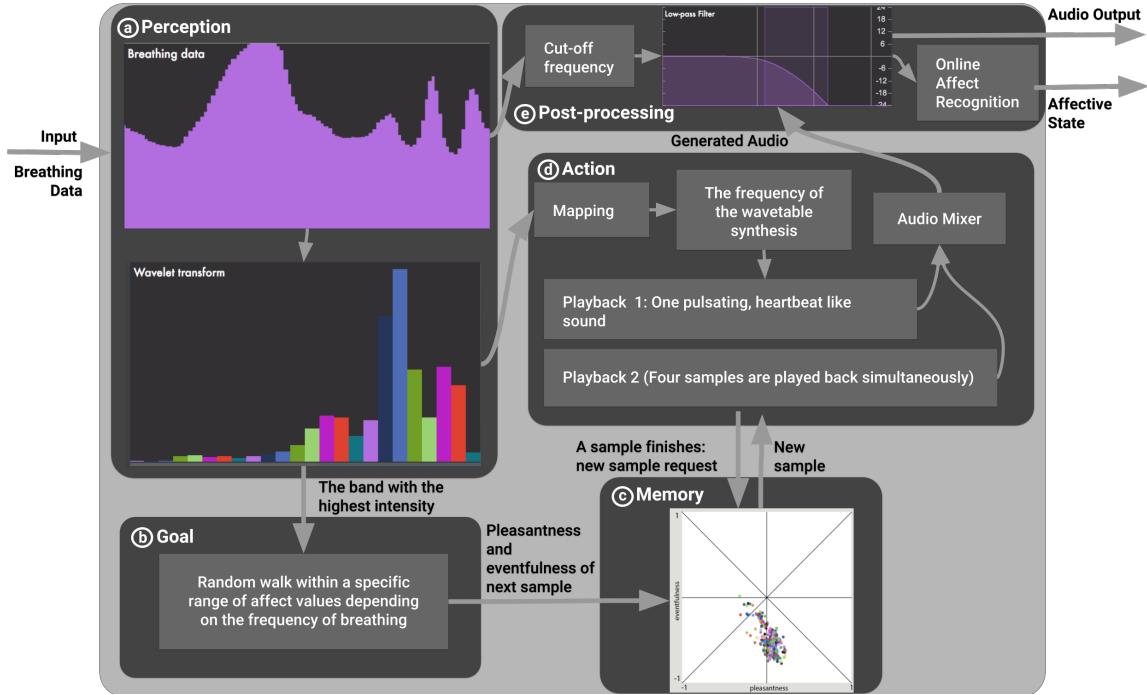


Figure 6.4: The musical agent in *Respire*

ent sounds for the musical agent's memory. We focused on quartal and quintal harmonies in piano recordings. The quartal and quintal harmony theory use the musical interval of fourths and fifths, thus generating ambient sounds that avoids tensions and resolutions of the tonal harmony. In line with the aesthetics of visuals, this harmonic choice evades imposing an explicit narrative on user. To obscure the source material of recordings, we suppressed the initial attacks of sounds using long fade-in durations. Then, we applied time stretch and pitch shifting to increase the number of samples in the sound memory. For the pitch-shifting, we applied intervals of fourths and fifths to stay within the quartal harmony. Then, we automatically labeled each sound using an affect estimation algorithm for sound. The labels are vectors with two dimensions: average pleasantness and average eventfulness of an audio sample. We previously published the details of this affect estimation algorithm where the multivariate linear regression model is trained on a ground-truth data [7]. The negative correlation between the eventfulness and pleasantness in Figure 6.3 has been previously observed in our previous studies. In sound studies, the affective dimensions, valence and arousal are exchanged with eventfulness and pleasantness because a sound doesn't feel an emotion, but it stimulates one. For example, there is no concept of a happy sound (excluding anthropomorphism), but some sounds trigger happiness emotions on humans.

6.5.2 Signal Processing of Breathing Data

The *perception* module recognizes the frequency of user's breathing by the wavelet transform of the breathing amplitude stream (Figure 6.4a.). The wavelet transform outputs the spectrum of a signal.

The breathing frequencies can go as low as 0.03 Hz. As the window size increases, we introduce longer delays to the system. The wavelet transform addresses this by using different window sizes to calculate each band, which provides for the detection of sudden changes in the signal while calculating low frequencies. In our implementation, the wavelet minimum frequency is 0.03 Hz, the maximum frequency is 2 Hz, the carrier frequency is 0.06 Hz, and there are 4 bands per octave. Hence, the output of the wavelet transform is the power of 24 bands. Lastly, the perception module of agent outputs the band with the highest power.

6.5.3 Generative Algorithm of Musical Agent

The goal module (Figure 6.4b) generates a vector with two dimensions, eventfulness and pleasantness, to select a sound from the memory (Figure 6.4c) to be played by the action module (Figure 6.4d). The memory module chooses the audio sample with the closest Euclidean distance to the 2D vector (pleasantness and eventfulness) generated by the goal module. The goal module generates the eventfulness values by using a mapping between the wavelet frequency with the highest power and the eventfulness of audio samples. The lowest and highest wavelet frequency bands are mapped to the lowest and highest eventfulness values of the agent's memory, respectively. Using these maximum values (Fig. 3), the frequency bands of breathing are mapped to 24 eventfulness values with equal distance. The agent applies a 2D random walk around one of these 24 eventfulness values so that the 24 wavelet bands correspond to 24 different areas in 2D affective space in Figure 6.3.

The action module incorporates two playback engines (Figure 6.4d). The first engine applies wavetable synthesis to generate a heartbeat like sound. The prominent frequency of breathing data is mapped to the looping frequency of the wavetable synthesis. Hence, the pulsation sound in the sonic environment slows down as the user breathes slower. Similarly, the pulsation speeds up following the acceleration in the breathing. The lower frequency bands slow down the pulsation sound to a point that the pulsation morphs into an ambient, pad-like sound. The second playback engine in the action module includes four voices. Each voice is active at all the times. When the goal module chooses an audio sample, this playback engine plays the given sample. When a sample finishes, the action module requests a new sample from the goal and memory module.

We developed the post-processing module to further enhance the interaction between the user and the audio environment by introducing a low-pass filter (Figure 6.4e). The amplitude of breathing controls the cut-off frequency of this filter. The cut-off frequency increases as the user breathes in, and vice versa. This mapping resembles the relation between musical pitch and movement [15]. Hence, as the user breathes in and out, timbre of the sonic environment oscillates between a muddy, low-frequency prominent audio environment and a full-spectrum audio environment. This mapping aims to enhance the submersion feeling of user.

Lastly, the agent applies an affect estimation algorithm to estimate the eventfulness and pleasantness of the generated audio environment (Figure 6.4e). The estimation algorithm is the online version of the estimation algorithm that is used to label the audio in the agent's memory. The output of the online affect estimation is a vector with two dimensions: eventfulness and pleasantness. These



Figure 6.5: A dancer in still position in P.O.E.M.A. (Photo: Adriano Fagundes.)

values are further used to control the parameters of the virtual reality environment (see Section 6.5 and Figure 6.2).

6.6 Exhibitions

Respire is the continuation of our previous artwork titled *Pulse.Breath.Water*. Both artworks share the same system design and we improved the visual environment in *Respire* by exploring different color schemes and virtual lights, and adding fogs to improve the ambiguous aesthetics. We presented *Pulse.Breath.Water* (PBW) within three collective exhibitions. The first exhibition named Scores + Traces: exposing the body through computation took place at the One Art Space gallery in Manhattan, New York, NY, USA in March 2016. The theme of the exhibition was movement and computation, and the exhibition brought a new perspective on how to incorporate movement theories in computational arts.

After the Scores + Traces exhibition in New York, we were invited by the Regina Miranda & Actors/Dancers Company for a collaboration to create and produce a piece, titled P.O.E.M.A (Percurso Organizados Entre Movimentos Aleatórios/ In English: Organized Paths Among Aleatory Movements), for the cultural program at the Rio Olympics 2016, which lasted for five weeks in Summer 2016.



Figure 6.6: A user with VR headset, a dancer, and spectators in the exhibition space of P.O.E.M.A. (Photo: Adriano Fagundes.)

P.O.E.M.A is a choreographic installation that incorporates contemporary dance to PBW (see Figure 6.5, 6.6, and 6.7). The piece was exhibited in a 10m by 10m room and the main challenge was creating a space that would bring both virtual environment of PBW and the dancers together. There were several spectators in the space while one audience member was in the virtual environment (Figure 6.6). The view of user in the virtual environment was projected on one wall. To expand the 2D projection of the virtual environment to the 3D space of dancers, we utilized a white colored space with a light design that was inspired by the work of James Turrell (Figure 6.5). This light design blended the 2D projection of the virtual environment to the 3D space of dancers. The audio followed this approach of blending, and we expanded the stereo output of the sonic environment to quadraphonics by using spectral spatialization.

Three dancers joined the project while one was performing at a time. The dancers were changing daily to have a rest after several hours of performance. The dancers interacted with virtual environment using a set of 200 choreographic cells, snippets of movements. These choreographic cells were the vocabulary of the dancers to react to the emerging behaviors that are initiated by the interaction between the user and the virtual environment. While the user in the virtual environment was changing; the visuals went to black, the lights were dimmed, the dancers were in a static posture, and there were two video loops of three dancers on the side walls.

The third exhibition of Pulse.Breath.Water, the MUTEK Mixed Realities VR Exhibition (November 2016) included forty virtual reality artworks covering interactive immersive environments, narrations within immersive environments, and 360 videos [1]. We heard that Pulse.Breath.Water brought a new perspective to the exhibition by provoking the idea of exploring a virtual presence



Figure 6.7: A dancer in movement in P.O.E.M.A. (Photo: Adriano Fagundes.)

using breathing as a way of interacting with the immersive environment. Following, we revisited the visuals and created Respire. We exhibited this new version at the CHI 2018 VR exhibition in Montreal Canada in April 2018 [14], and Digital Carnival 2018 in Vancouver Canada in August 2018 [2].

6.7 Next Steps

The overall interaction design and the system architecture of Respire stand closer to reactivity in comparison to interactivity. Tatar and Pasquier [17] clarify that interactivity of agents involves proactivity, that is planning future actions, and interacting with other human, biological, or artificial agents. A next step could be to research how the introduction of proactive behaviors would affect the ambiguity and meaning making of Respire's user experience. For example, in an interactive scenario, periods of prolonged fast breathing could lead to a calming of the water surface and less musical agent activity which in turn may lead to a change of breathing of the user. However, our previous exhibition experiences showed that the audience interacted with Respire in mainly two phases: a playful phase where the effect of the breathing and the sensor was explored, followed by a calmer phase where the user started moving less than the initial phase. Although these findings are speculative, they initiate a direction for further research on reactive and interactive scenarios.

Bibliography

- [1] Artworks | MUTEK, . URL <http://www.mutek.org/en/img/2016/artworks>.
- [2] Digital Carnival Twines Ancestral Wisdom and Digital Innovation Aug 31 & Sep 1, 2018 - Cinevolution, . URL <http://cinevolutionmedia.com/digital-carnival-twines-ancestral-wisdom-and-digital-innovation-aug-31-sep-1-2018/>.
- [3] M+M Middleware, . URL <https://hplustech.com/blogs/news/m-m-middleware>.
- [4] Respire, . URL <https://kivanctatar.com/Respire>.
- [5] Char Davies. OSMOSE: Notes on being in Immersive virtual space. *Digital Creativity*, 9(2): 65–74, January 1998. ISSN 1462-6268, 1744-3806. doi: 10.1080/14626269808567111. URL <http://www.tandfonline.com/doi/abs/10.1080/14626269808567111>.
- [6] Paul Dourish. *Where the Action Is: The Foundations of Embodied Interaction*. The MIT Press, Cambridge, Mass., 1 edition edition, August 2004. ISBN 978-0-262-54178-7.
- [7] Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society*, 64(9): 646–653, September 2016. ISSN 15494950. doi: 10.17743/jaes.2016.0044. URL <http://www.aes.org/e-lib/browse.cfm?elib=18373>.
- [8] William W Gaver, Jacob Beaver, and Steve Benford. Ambiguity as a Resource for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*, pages 233–240, NY, USA, 2003. ACM.
- [9] Eric Kandel. *Reductionism in Art and Brain Science: Bridging the Two Cultures*. Columbia University Press, New York, 1 edition edition, August 2016. ISBN 978-0-231-17962-1.
- [10] NIMkartchannel. Sonia Cillari: As an Artist, I Need to Rest (2009). URL <https://www.youtube.com/watch?v=GcA6Yw2QERo>.
- [11] Jaak Panksepp. *Affective Neuroscience: The Foundations of Human and Animal Emotions*. Oxford University Press, September 1998. ISBN 978-0-19-802567-2.
- [12] Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An Introduction to Musical Metacreation. *Computers in Entertainment*, 14(2):1–14, January 2017. ISSN 15443574. doi: 10.1145/2930672. URL <http://dl.acm.org/citation.cfm?doid=3023311.2930672>.

- [13] Mirjana Prpa, Kivanç Tatar, Bernhard E. Riecke, and Philippe Pasquier. The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality. In *Virtual, Augmented and Mixed Reality, 9th International Conference, VAMR 2017, Held as Part of HCI International 2017, Proceedings*, Vancouver, 2017. Springer. ISBN 978-3-319-57986-3. URL <http://www.springer.com/gp/book/9783319579863>.
- [14] Mirjana Prpa, Thecla Schiphorst, Kivanç Tatar, and Philippe Pasquier. Respire: a Breath Away from the Experience in Virtual Environment. In *CHI EA '18 Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–6. ACM Press, 2018. ISBN 978-1-4503-5621-3. doi: 10.1145/3170427.3180282. URL <http://dl.acm.org/citation.cfm?doid=3170427.3180282>.
- [15] B. Sievers, L. Polansky, M. Casey, and T. Wheatley. Music and movement share a dynamic structure that supports universal expressions of emotion. *Proceedings of the National Academy of Sciences*, 110(1):70–75, January 2013. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1209023110. URL <http://www.pnas.org/cgi/doi/10.1073/pnas.1209023110>.
- [16] Kivanç Tatar and Philippe Pasquier. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, June 2017. ISBN 978-1-77287-019-0.
- [17] Kivanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):1–50, September 2018. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1511736. URL <https://www.tandfonline.com/doi/full/10.1080/09298215.2018.1511736>.
- [18] Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, June 2009. ISBN 978-0-470-51946-2.

Chapter 7

Summary and Conclusion

7.1 Summary

Metacreation is an interdisciplinary field where the scientific knowledge of Computational Creativity is put into the artistic practice of Generative and Interactive Arts. Metacreation and Musical Metacreation (MuMe) explore the notion of meta-level creativity in which the artist creates an autonomous system that makes artworks, in our case, music. The products of metacreative process are generative in nature, powered with autonomous behaviours. In addition to the generative and autonomous behaviours, musical agents exhibit reactivity, proactivity, interactivity, adaptability, versatility, and volition by utilizing the advanced technologies of Multi-agent Systems, Machine Learning, and Artificial Intelligence.

This thesis started by introducing the Generative arts and Computational Creativity while clarifying the notions of creativity, style, and style imitation. The initial contribution of this thesis is the state of the art of musical agent architectures in Chapter 2. We surveyed 78 musical agent systems and proposed a typology of musical agents in nine dimensions of musical tasks, environment types, number of agents, number of agent roles, communication types, corpus types, input/output types, and human interaction modality (HIM). Chapter 2 continued by giving the details of 78 musical agent architectures while grouping these systems by their architecture type. Section 2.7 informed the reader on the evaluation methodologies of musical agents while pointing out possible benchmarking opportunities for musical agent research. Section 2.8 remarked the interdisciplinarity of MuMe while indicating possible future steps in MuMe studies.

The survey of musical agents indicated a possible research direction of audio-based musical agents with unsupervised learning. Chapter 3 introduced a musical agent architecture called Musical Agents based on Self-Organizing Maps, (MASOM). The musical context of this agent architecture is experimental electronic music including (but not limited to) electroacoustic music, mainstream electronic music, glitch and noise music, and Intelligent Dance Music (IDM). MASOM's architecture builds on the electroacoustic and contemporary music theories that define music as "nothing but organized sounds." In that regards, we combined sound memory organization with temporal musical structure modelling in MASOM's architecture. The sounds memory organization utilizes the neural networks model, Self-Organizing Maps to automatically create a latent space for audio samples. The temporal structure modelling focus on statistical sequence modelling to organize sounds in time.

We tested four statistical sequence modelling algorithms within the MASOM architecture: Factor Oracle, Recurrent Neural Networks, Variable Markov Models Max-order variation, and Variable Markov Model Prediction by Partial Matching C variation. Chapter 4 clarified the system details of these variations in the architecture of MASOM, and compares these four algorithms for temporal musical structure modelling in MASOM. Our analysis proposed three analytical measures: n-gram cloning, repetition analysis, and the longest subsequence analysis. N-gram cloning measures the degree to which the generated sequences clone sub-sequences from the original data. The repetition analysis indicates the variance of sub-sequences within each generated sequence. The longest subsequence cloning shows the length of the longest subsequence that is directly copied from the training

corpus. These three analytical measures provided a descriptive understanding of the sequence generated by these models. We argued that these measures move beyond the conventional understanding of optimality in Machine Learning studies by providing a comprehension of how model works in musically meaningful ways. As the Computational Creativity and Musical Metacreation studies suggested [17, 1], the notion of optimality is ill-defined in music and creative applications in general. Hence, we concluded with an understanding of how these statistical sequence models work in our creative applications.

Building on the descriptive knowledge that we created in Chapter 4, we applied the MASOM Factor Oracle variation to the creative application of live audio-visual performance. Chapter 5 introduced the project *Revive*, an audio-visual live performance project including three sonic performers: Kivanç Tatar, Philippe Pasquier, and MASOM. The *Revive* project concentrates on structured improvisation using an automatized cue system. The cues define aesthetic constraints of sonic performers within a musical section. The visuals of *Revive* is projected on a dome surface, and the audio is spatialized using a 3D audio setup with 157 speakers clustered to 31 channels. The project also includes three audio-reactive visual agents that emphasizes the action of sonic performers using audio-reactive generative visuals. The *Revive* further enhances the audience comprehension of the correlation between the sonic gestures and generative visuals by spatializing the sonic gestures where the visuals appear. This synchronization of visuals and sounds applies 3D audio spatialization techniques including strategies for generative spatial trajectories and interfaces for 3D audio spatialization.

Chapter 6 presented another creative application of an audio-based musical agent architecture. The project *Respire* focuses on using the user's breathing patterns as a way of interaction with a Virtual Reality (VR) environment. The musical agent that we explained in this chapter perceives the user's breathing patterns using signal processing. The agent maps the prominent frequency of the breathing to the eventfulness of audio sample selection. Hence, the *Respire* aims to react to the user's breathing where the generative music becomes more eventful when the user breaths faster, and vice versa. We clarified the difference between reactivity and interactivity in the context of musical agents in Chapter 2. Using this knowledge, we clarified the reactive behaviours of the musical agent in the *Respire*, and propose future research directions to study possible interactive musical agent behaviors for the specific application of the *Respire*.

7.2 Limitations

In Chapter 2 Section 2.8.2, we emphasize the strong connection between the musical agent studies and Music Information Retrieval (MIR) while explaining the future steps of research on musical agents. Electroacoustic music theories clarify that music composition and musical performance involve multiple concurrent musical layers[15, 14, 12, 13]. The current version of MASOM architecture is single-layer; that is, the agent learns as if there exists only one agent performing in the audio recordings included in the training set. That is, the learning procedure of MASOM also pre-

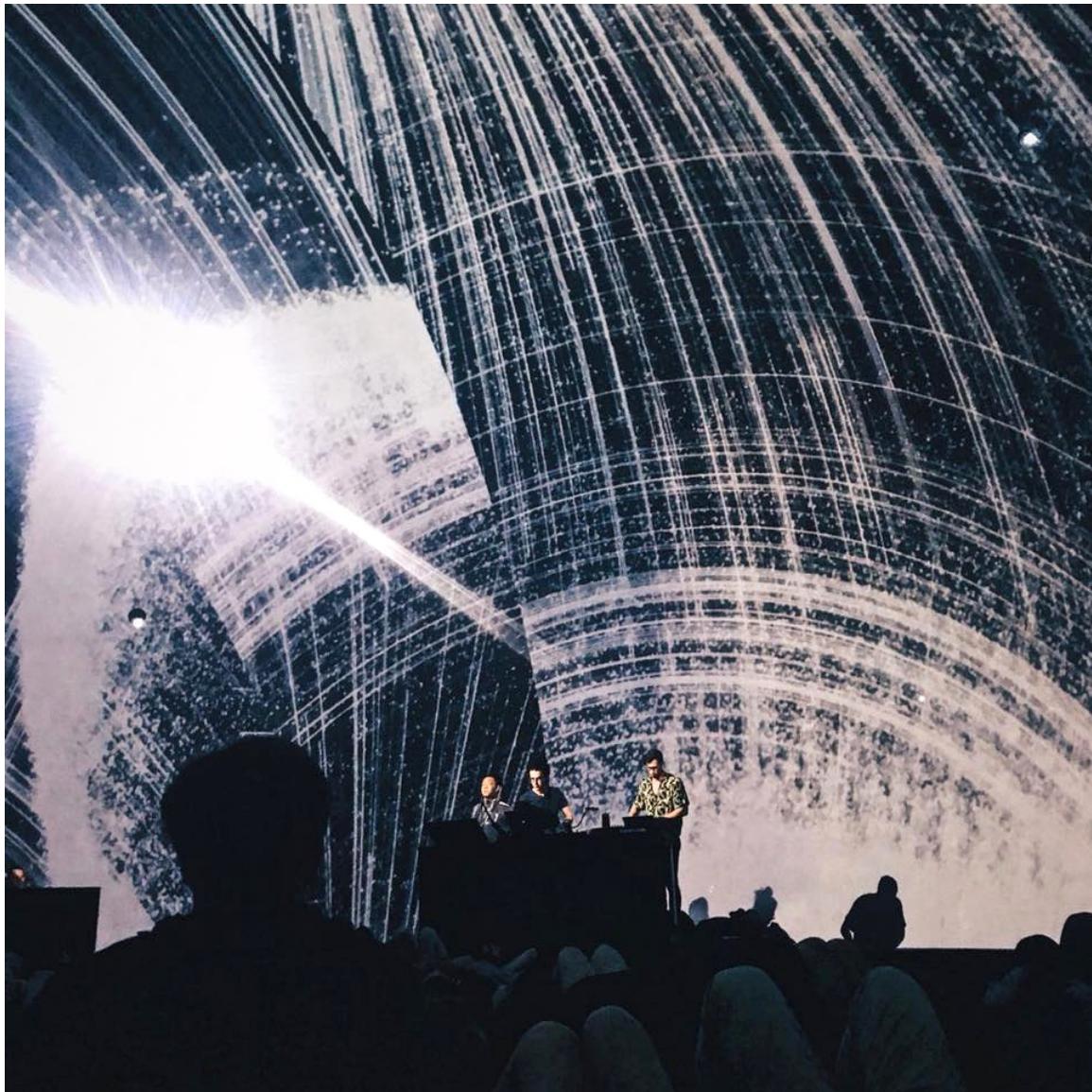


Figure 7.1: Revive performance at the Société des Arts Technologique during the festival MUTEK Montreal 2018, photo credit: Ashley Gesner



Figure 7.2: Revive members from left to right, Remy Siu, Kivanç Tatar, and Philippe Pasquier, photo credit: Ashley Gesner

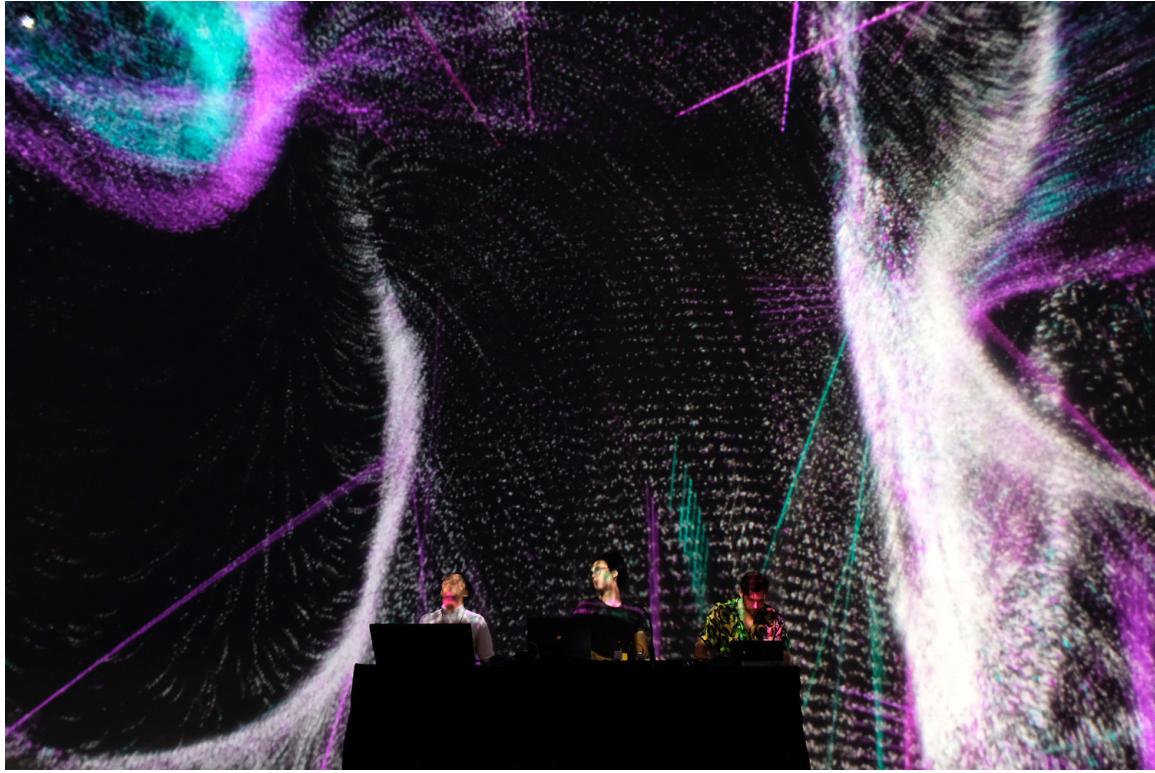


Figure 7.3: The audio-reactive particle engine in Revive's visuals, photo credit: Ashley Gesner

sumes that the recording is a single layer musical composition. This limitation is mainly because of the lack of multi-channel corpus of experimental electronic music, and the audio recordings are mainly shared as two channel stereo files. A possible solution for this issue is automatically separating individual layers (or stems in audio production terminology) from stereo recordings. However, automatic source separation for audio is still an open research area in MIR studies [10, 11]. In addition, the statistical sequence learning models in Machine Learning literature are mainly single-layer. The polyphonic music generation studies concentrate on multi-layer sequence generation for musical structure modelling; however, it is still an ongoing research area. In that regards, we propose possible future research directions in Section 7.3.2.

The studies presented in this thesis have not exploited all possible research questions on variations of machine listening models for audio-based musical agents. Tuuri and Eerola [22] proposed the revised taxonomy of listening where the authors build on the previous listening taxonomies by proposing three high level modes of listening: experiential, denotative, and reflective. A possible research direction for the research on MASOM is incorporating these listening modes into the musical agent architecture.

Section 2.7 attempts to propose a typology of evaluation methodologies for musical agent studies. While this attempt initiates a discussion on the topic, the proposed typology has the following limitations. Although the typology brings together the evaluations types proposed in the literature, the combined typology is not mutually exclusive, and this introduces confusion in the communica-

tion of this typology. For example, we differentiate three types of formal evaluation methodologies: 1- peer reviewers, curators and jury; 2- theoretical and analytic measures; 3- empirical studies. However, the evaluation methodology in Yee-King and d'Inverno [25] can be labeled by all three categories of our formal evaluations types. In our future work, we plan to revisit the evaluation typology and introduce categories that are mutually exclusive.

Our evaluation in Chapter 4 involves an analytical evaluation where we propose three measure to evaluate musical agents. The discussion on the evaluation methodologies of autonomous systems has been ongoing in the field (see 2). In the Computational Creativity literature, there has been attempts to define evaluation dimensions, measures, and methodologies for autonomous systems [9]. However, these proposal are still mostly theoretical and the procedures for practical implementations of these methodologies are yet to be given.

The artworks generated using MASOM system have been evaluated by the curators of well-known international festivals in the selection process. In Chapter 4, we examine the musical structure generation in MASOM using three proposed analytical measures of n-gram cloning, the longest sub-sequence cloning, and repetition. In addition, our future work includes conducting empirical evaluation experiments with the users of MASOM, and the audience members listening to the content generated using MASOM.

In the following, we build on these limitations by proposing future steps in our research on audio-based musical agents with unsupervised learning.

7.3 Future Work

We approach Metacreation and Musical Metacreation as interdisciplinary fields that bring scientific literature and artistic applications together. We aimed that our typology of musical agents creates a strong foundation for these disciplines. The musical agent architectures presented in this thesis exemplified the strengths of applying electroacoustic music theories in audio-based musical agent architectures with unsupervised learning. We believe that the notion of combining automatic audio latent space generation for sound memory organization with statistical sequence models for musical structures indicated new research directions. In that regards, we hope that the work in this thesis created a basis for three main research directions that would require attention in the following years: automatic latent space generation for audio, hierarchical models for audio-based musical agents, and embodied musical agents.

7.3.1 Automatic Latent Space Generation for Audio

The sound memory organization of musical agents that we presented in this thesis utilizes automatic latent space generation for audio. Similar sounds locate closer to each other in the latent audio space. Creating latent spaces of audio recordings elicits human-readable representation of audio samples for compositional tasks. This approach resonates with the second wave of HCI, which proposes the introduction of human cognition qualities to interactions with computers [6]. Moreover, artists have

been using the latent audio space during the composition process in various ways. Three examples of applications of latent audio space in the compositional practice are 1- constraining the timbre space per composition or per musical section, 2- creating contrast by introducing complementary timbres to a musical section, 3- generating sonic variety in finely constrained timbre spaces.

There have been attempts of latent audio space generation, which dates back to 1970s. Grey [8] studied how to create a latent space of audio samples that imitates 16 acoustic instruments. Toivainen et al. [21] studied the effect of timbre similarity in brainwave recordings, behavioral studies and computational simulations using Self-Organizing Maps. The researchers concluded that timbre similarity measured in behavioral studies corresponded to neural activity and activity in simulations of SOMs, indicating that SOMs may be used for modelling latent sonic spaces. Following, Eigenfeldt and Pasquier [3] proposed SOMs for selecting samples from a user-generated latent audio space. The generation of the latent space uses an audio thumbnailing with three dimensions. These dimensions are audio features selected by the user. Although this approach was promising, the user selected audio feature representations and the low number of dimensions do not guarantee a latent audio space generation that correlates to cognitive aspects of listening. Following Eigenfeldt and Pasquier's work, Fried et al. [7] apply the kernel-based sorting for fixed-sized, discrete latent audio space generation. The application of this study is limited to musical interfaces with a fixed number of buttons that correspond to the kernels in the model.

The work presented in this thesis applied SOMs for autonomous latent audio space generation for musical agents [16, 20]. Previously, we applied SOMs for organizing the timbre memory of musical agents. The audio thumbnailing of our previous approach combined well-known audio features with affective dimensions of human cognition [4, 5]. We also researched computational approaches to calculate sound similarity for various musical tasks such as automatic synthesizer preset generation [18] and interactive music systems for machine improvisation [16], live performances [19], and virtual reality applications (see Chapter 6). These studies indicate a research direction on how to use autonomous latent audio spaces for MuMe applications.

This doctoral study started in September 2014 that corresponds to the beginning of the advancements in Deep Learning. During this time period, Deep Learning moved advanced exponentially and many more models have been proposed for multimedia applications. Building on our studies mentioned above, we aim to explore and compare several ML and AI algorithms of Neural Networks, Deep Learning models, and hierarchical pattern recognition models for the autonomous generation of latent audio spaces.

7.3.2 Hierarchical Models for Audio-based Musical Agents

Our research on Musical Agents based on Self-Organizing Maps indicated an application of hierarchical models within a musical agent architecture. A hierarchical model applied to autonomous latent audio space generation can provide representation with a balance of granularity and generality. In MASOM, the single layer audio latent space is flexible to be trained on any audio corpus. We observed that the single layer representation provides a high level of granularity. However, a

latent audio space representation with a high granularity complicates the statistical sequence model training, and such audio representation can lead to an increased dimensionality. We think that a new research direction where a hierarchical latent space representation combined with a hierarchical statistical sequence model can introduce a balance where the agent learns audio differentiation with a high granularity and temporal musical structure with generality.

7.3.3 Automatic Spatialization of Musical Agents

Spatialization of sonic gestures requires audio rate movement trajectories. Generating such movement trajectories introduces more parameters to a performance. Hence, automatizing the spatialization of sonic gestures can help performers to deal with the complexity of 3D audio. Previously, we implemented autonomous spatial audio motion trajectory generation using three strategies: circular, tangential, and random-walking (see Chapter 5). Exciting research possibilities still exist on how to generate spatial trajectories that are correlated with the sonic output of a musical agent. A system for automatization of spatialization could consist of four modules: detection of sonic gestures using onset detection and musical structure analysis, calculating audio features per sonic gesture, using these features to generate possible spatial movement trajectories, updating these trajectories based on the spatial locations and the timbre of other agents in the environment.

7.3.4 Visualization and Embodiment Musical Agents

Musical gestures are often lost in electronic music performances. This problem also appears in machine improvisation performances. In computer music performances with multiple performers, it may not be clear which agent is generating a specific sonic gesture. To overcome this problem, visualization of musical gestures can improve the audience comprehension of the connection between a sonic gesture and the agent generating the gesture. Previous studies showed that visual cues help the identification of musical gestures [23, 24].

Designing visual agents to visualize the musical gestures and actions of sonic agents can help bringing back the spectacle aspect of performance to machine improvisation. Proposed visual agent architecture would be autonomous and the visual agents would react to the decisions made by sonic agents. Visual agents can perceive the action of sonic agents using a machine listening algorithm with high-level music features such as eventfulness and pleasantness as well as low-level features such as spectral features, loudness, and pitch related features. The visual agents can also perceive if an audio agent is active and when an audio agent initiates sound. Using perception abilities, the visual agents emphasize the actions of audio agents and make it easier for the audience to comprehend the connection between sonic gestures and the sonic agents' actions.

Bretan and Weinberg [2] inform the current state of the art of robotic musicianship. The intersection of robotics and musical agents can provide exciting opportunities for the embodiment of musical agents. The survey of Bretan and Weinberg indicates that new research possibilities exist in the cross-section of audio-based musical agents and robotics in the context of experimental elec-

tronic music. Moreover, Virtual agents in Computer-Generated Imagery (CGI) agent can be further explored to embody musical agents in virtual environments.

7.4 Final Word

The tools of music production influence how we create music, and expand our understanding of what music is. Music technology is in strong connection with how we make music. Musical Metacreation explores new autonomous tools provided by the technologies of Machine Learning and Artificial Intelligence for musical creativity. We hope that the interdisciplinary research of MuMe and the tools created by MuMe studies shape how we create music in the future while introducing new concepts that surround music.

Bibliography

- [1] Margaret A. Boden. Creativity and ALife. *Artificial Life*, 21(3):354–365, August 2015. ISSN 1064-5462, 1530-9185. doi: 10.1162/ARTL_a_00176. URL http://www.mitpressjournals.org/doi/10.1162/ARTL_a_00176.
- [2] Mason Bretan and Gil Weinberg. A survey of robotic musicianship. *Communications of the ACM*, 59(5):100–109, April 2016. ISSN 00010782. doi: 10.1145/2818994. URL <http://dl.acm.org/citation.cfm?doid=2930840.2818994>.
- [3] Arne Eigenfeldt and Philippe Pasquier. Real-Time Timbral Organisation: Selecting samples based upon similarity. *Organised Sound*, 15(02):159–166, August 2010. ISSN 1469-8153. doi: 10.1017/S1355771810000154. URL http://journals.cambridge.org/article_S1355771810000154.
- [4] Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society*, 64(9):646–653, September 2016. ISSN 15494950. doi: 10.17743/jaes.2016.0044. URL <http://www.aes.org/e-lib/browse.cfm?elib=18373>.
- [5] Jianyu Fan, Kivanç Tatar, Miles Thorogood, and Philippe Pasquier. Ranking-Based Emotion Recognition for Experimental Music. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) 2017*, 2017.
- [6] Michael Filimowicz. *New directions in third wave human-computer interaction: volume 1 - technologies*. Springer Berlin Heidelberg, New York, NY, 2018. ISBN 978-3-319-73355-5.
- [7] Ohad Fried, Zen Jin, and Reid Oda. AudioQuilt: 2d Arrangements of Audio Samples using Metric Learning and Kernelized Sorting. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths University of London, 2014. URL <http://www.nime2014.org/technical-programme/proceedings/>.

- [8] John M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, May 1977. ISSN 0001-4966. doi: 10.1121/1.381428. URL <http://scitation.aip.org.proxy.lib.sfu.ca/content/asa/journal/jasa/61/5/10.1121/1.381428>.
- [9] Anna Jordanous. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation*, 4(3):246–279, September 2012. ISSN 1866-9956, 1866-9964. doi: 10.1007/s12559-012-9156-1. URL <http://link.springer.com/10.1007/s12559-012-9156-1>.
- [10] Ethan Manilow, Prem Seetharaman, and Bryan Pardo. The northwestern university source separation library. Proceedings of the 19th International Society of Music Information Retrieval Conference (ISMIR 2018), Paris, France, September 23-27, 2018.
- [11] Ethan Manilow, Prem Seetharaman, Fatemeh Pishdadian, and Bryan Pardo. NUSSL: the northwestern university source separation library. <https://github.com/interactiveaudiolab/nussl>, 2018.
- [12] Curtis Roads. *Microsound*. The MIT Press, Cambridge, Mass., August 2004. ISBN 9780262681544.
- [13] Curtis Roads. *Composing electronic music: a new aesthetic*. Oxford University Press, Oxford, 2015. ISBN 978-0-19-537324-0.
- [14] Denis Smalley. Spectromorphology: explaining sound-shapes. *Organised Sound*, 2(02):107–126, August 1997. ISSN 1469-8153. doi: 10.1017/S1355771897009059. URL http://journals.cambridge.org/article_S1355771897009059.
- [15] Karlheinz Stockhausen. Four Criteria of Electronic Music with Examples from Kontakte, 1972. URL <https://www.youtube.com/watch?v=7xyGtI7KKIY&list=PLRBdTyZ761vAFotZvocPjpRVTL6htJzoP>.
- [16] Kivanç Tatar and Philippe Pasquier. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, June 2017. ISBN 978-1-77287-019-0.
- [17] Kivanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):1–50, September 2018. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1511736.
- [18] Kivanç Tatar, Matthieu Macret, and Philippe Pasquier. Automatic Synthesizer Preset Generation with PresetGen. *Journal of New Music Research*, 45(2):124–144, April 2016. ISSN 0929-8215. doi: 10.1080/09298215.2016.1175481. URL <http://dx.doi.org/10.1080/09298215.2016.1175481>.

- [19] Kivanç Tatar, Philippe Pasquier, and Remy Siu. REVIVE: An Audio-visual Performance with Musical and Visual AI Agents. pages 1–6. ACM Press, 2018. ISBN 978-1-4503-5621-3. doi: 10.1145/3170427.3177771. URL <http://dl.acm.org/citation.cfm?doid=3170427.3177771>.
- [20] Kivanç Tatar, Philippe Pasquier, and Remy Siu. Audio-based Musical Artificial Intelligence and Audio-Reactive Visual Agents in Revive. New York, NY, US, 2019. International Computer Music Association.
- [21] Petri Toivainen, Mauri Kaipainen, and Jukka Louhivuori. Musical timbre: Similarity ratings correlate with computational feature space distances*. *Journal of New Music Research*, 24(3):282–298, September 1995. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298219508570686. URL <http://www.tandfonline.com/doi/abs/10.1080/09298219508570686>.
- [22] Kai Tuuri and Tuomas Eerola. Formulating a Revised Taxonomy for Modes of Listening. *Journal of New Music Research*, 41(2):137–152, June 2012. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2011.614951. URL <http://www.tandfonline.com/doi/abs/10.1080/09298215.2011.614951>.
- [23] Marcelo M. Wanderley. Quantitative Analysis of Non-obvious Performer Gestures. In G. Goos, J. Hartmanis, J. van Leeuwen, Ipke Wachsmuth, and Timo Sowa, editors, *Gesture and Sign Language in Human-Computer Interaction*, volume 2298, pages 241–253. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-540-43678-2 978-3-540-47873-7. doi: 10.1007/3-540-47873-6_26. URL http://link.springer.com/10.1007/3-540-47873-6_26.
- [24] Marcelo M Wanderley, Bradley W Vines, Neil Middleton, Cory McKay, and Wesley Hatch. The Musical Significance of Clarinetists' Ancillary Gestures: An Exploration of the Field. *Journal of New Music Research*, 34(1):97–113, March 2005. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298210500124208. URL <http://www.tandfonline.com/doi/abs/10.1080/09298210500124208>.
- [25] Matthew Yee-King and Mark d'Inverno. Experience driven design of creative systems. In *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016). Université Pierre et Marie Curie*, 2016. URL <http://www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/Experience-Driven-Design-of-Creative-Systems.pdf>.

Cumulative Bibliography

Oxford Dictionaries English. Oxford University Press, 2017. URL <https://en.oxforddictionaries.com/definition/style>.

Asmaa Majid Al-Rifaie and Mohammad Majid Al-Rifaie. Generative Music with Stochastic Diffusion Search. In Colin Johnson, Adrian Carballal, and João Correia, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, number 9027 in Lecture Notes in Computer Science, pages 1–14. Springer International Publishing, April 2015. ISBN 978-3-319-16497-7 978-3-319-16498-4. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-319-16498-4_1.

Alo Allik. Gene expression synthesis. In *the Proceedings of the Joint Conference ICMC14-SMC14*, Athens, Greece, 2014. URL http://tehis.net/acquire/AloAllik_GeneExpressionSynthesis.pdf.

Alvaro Ámorim, Luís Fabrício W. Góes, Alysson Ribeiro da Silva, and Celso Francsá. Creative Flavor Pairing: Using RDC Metric to Generate and Assess Ingredients Combinations. In *Proceedings of the Eight International Conference on Computational Creativity (ICCC 2017)*, 2017. URL http://computationalcreativity.net/iccc2017/ICCC_17_accepted_submissions/ICCC-17_paper_40.pdf.

Daichi Ando and Hitoshi Iba. Real-time Musical Interaction between Musician and Multi-agent System. In *Proceedings of the 8th Generative Art Conference*, 2005. URL <http://www.iba.t.u-tokyo.ac.jp/papers/2005/dandoGA2005.pdf>.

I Arel, D C Rose, and T P Karnowski. Deep Machine Learning - A New Frontier in Artificial Intelligence Research. *IEEE Computational Intelligence Magazine*, 5(4):13–18, November 2010. ISSN 1556-603X. doi: 10.1109/MCI.2010.938364. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5605630>.

Kat Arges, Jamie Forth, and Geraint A. Wiggins. Evaluation of musical creativity and musical metacreation systems. *Computers in Entertainment (CIE) - Special Issue on Musical Metacreation, Part II*, 14(3), 2016. URL <https://qmro.qmul.ac.uk/xmlui/handle/123456789/14230>.

Jaime Arias, Myriam Desainte-Catherine, and Shlomo Dubnov. Automatic Construction of Interactive Machine Improvisation Scenarios from Audio Recordings. In *The Fourth International Workshop on Musical Metacreation (MUME 2016)*, 2016. URL <https://hal.archives-ouvertes.fr/hal-01336825/>.

Gérard Assayag and Shlomo Dubnov. Using Factor Oracles for Machine Improvisation. *Soft Computing*, 8(9):604–610, August 2004. ISSN 1432-7643, 1433-7479. doi: 10.1007/s00500-004-0385-4. URL <http://link.springer.com.proxy.lib.sfu.ca/article/10.1007/s00500-004-0385-4>.

Gérard Assayag, Shlomo Dubnov, and Olivier Delerue. Guessing the Composer’s Mind: Applying Universal Prediction to Musical Style. In *Proceedings of the 1999 International Computer Music Conference, ICMC 1999*, page 6, Beijing, China, 1999.

Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. Omax brothers: a dynamic topology of agents for improvisation learning. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia*, pages 125–132. ACM Press, 2006. URL <http://dl.acm.org/citation.cfm?id=1178742>.

Jean-Julien Aucouturier. Artificial Evolution of Tuning Systems. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.

Jean-Julien Aucouturier and François Pachet. Ringomatic: A Real-Time Interactive Drummer Using Constraint-Satisfaction and Drum Sound Descriptors. In *In Proceedings of the International Conference on Music Information Retrieval*, pages 412–419, 2005. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.81.4681&rep=rep1&type=pdf>.

Pascal Baltazar, Théo De La Hogue, and Myriam Desainte-Catherine. Demo: i-score, an Interactive Sequencer for the Intermedia Arts. 2014. doi: 10.13140/2.1.3556.0008.

Ron Begleiter, Ran El-Yaniv, and Golan Yona. On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, 22:385–421, 2004. URL <http://www.jair.org/papers/paper1491.html>.

Francisco Bernardo, Michael Zbyszynski, Rebecca Fiebrink, and Mick Grierson. Interactive Machine Learning for End-User Innovation. In *Proceedings of AAAI Spring Symposium*. American Association for Artificial Intelligence (AAAI), 2016. URL <http://research.gold.ac.uk/id/eprint/19767>.

Frédéric Bevilacqua, Bruno Zamborlin, Anthony Sypniewski, Norbert Schnell, Fabrice Guédy, and Nicolas Rasamimanana. Continuous realtime gesture following and recognition. In *International gesture workshop*, pages 73–84. Springer, 2009.

Peter Beyls. Interaction and Self-organisation in a Society of Musical Agents. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, 2007. URL http://cmr.soc.plymouth.ac.uk/publications/musical_beyls.pdf.

Peter Beyls. On-line Development of Man-Machine Relationships: Motivation-driven Musical Interaction. In *Proceedings of the 11th Generative Art Conference*, Milan, Italy, 2008. URL <http://www.generativeart.com/on/cic/papersGA2008/1.pdf>.

Peter Beyls. Interactive Composing as the Expressions of Autonomous Machine Motivations. In *Proceedings of the International Computer Music Conference (ICMC 2009)*, Montreal, Canada, 2009.

Peter Beyls. Structural Coupling in a Society of Musical Agents. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.

Peter Beyls. Autonomy, Influence and Emergence in an Audiovisual Ecosystem. In *Proceedings of the Generative Arts Conference, Rome, Italy*, 2012. URL <http://www.generativeart.it/GA2012/peter.pdf>.

Peter Beyls, Gilberto Bernardes, and Marcelo Caetano. earGram Actors: An Interactive Audiovisual System Based on Social Behavior. *Journal of Science and Technology of the Arts*, 7(1):43–54, 2015. URL <http://artes.ucp.pt/citarj/article/download/142/104>.

John Biles. GenJam: A genetic algorithm for generating jazz solos. In *Proceedings of the International Computer Music Conference*, pages 131–131. International Computer Music Association, 1994. URL <http://igm.rit.edu/~jabics/GenJam94/Paper.html>.

John A. Biles. Performing with Technology: Lessons Learned from the GenJam Project. In *Proceedings of the 2nd International Workshop on Musical Metacreation (MUME 2013)*, Boston, MA, 2013. URL <http://musicalmetacreation.org/mume2013/content/proceedings/Performing%20with%20Technology-%20Lessons%20Learned%20from%20the%20GenJam%20Project.pdf>.

Jim Bizzocchi, Arne Eigenfeldt, and Miles Thorogood. Generating Affect: Applying Valence and Arousal values to unified video, music, and sound generation system. In *Proceedings of the 18th Generative Art Conference*, volume 49, pages 621–630, Venice, 2015. URL http://www.generativeart.com/ga2015_WEB/generating-affect_eigenfeldt.pdf.

Tim Blackwell and Michael Young. Self-organised music. *Organised Sound*, 9(02):123–136, 2004. URL http://journals.cambridge.org/abstract_S1355771804000214.

Tim Blackwell, Oliver Bown, and Michael Young. Live Algorithms: Towards Autonomous Computer Improvisers. In Jon McCormack and Mark d’Inverno, editors, *Computers and Creativity*, pages 147–174. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-31726-2 978-3-642-31727-9. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-642-31727-9_6.

Georges Bloch, Shlomo Dubnov, and Gérard Assayag. Introducing video features and spectral descriptors in the omax improvisation system. In *International Computer Music Conference ’08*, 2008. URL <https://hal.archives-ouvertes.fr/hal-01161405/>.

Margaret A Boden. Computer models of creativity. *AI Magazine*, 30(3):23, 2009.

Margaret A. Boden. Creativity and ALife. *Artificial Life*, 21(3):354–365, August 2015. ISSN 1064-5462, 1530-9185. doi: 10.1162/ARTL_a_00176. URL http://www.mitpressjournals.org/doi/10.1162/ARTL_a_00176.

Paul M Bodily and Dan Ventura. Musical Metacreation: Past, Present, and Future. In *Proceedings of the Sixth International Workshop on Musical Metacreation*, page 5, 2018.

Benjamin David Robert Bogart and Philippe Pasquier. Context machines: A series of situated and self-organizing artworks. *Leonardo*, 46(2):114–122, 2013. URL http://www.mitpressjournals.org/doi/abs/10.1162/LEON_a_00525.

O. Bown, J. McCormack, and T. Kowaliw. Ecosystemic methods for creative domains: Niche construction and boundary formation. In *2011 IEEE Symposium on Artificial Life (ALIFE)*, pages 132–139, April 2011. doi: 10.1109/ALIFE.2011.5954651.

Oliver Bown. Experiments in Modular Design for the Creative Composition of Live Algorithms. *Computer Music Journal*, 35(3):73–85, 2011. ISSN 1531-5169. URL https://muse-jhu-edu.proxy.lib.sfu.ca/journals/computer_music_journal/v035/35.3.bown.html.

Oliver Bown and Aengus Martin. Backgammon: Process-based Musical Explorations Using the Agent Designer. In *Proceedings of the 9th ACM Conference on Creativity & Cognition, C&C '13*, pages 390–391, New York, NY, USA, 2013. ACM Press. ISBN 978-1-4503-2150-1. doi: 10.1145/2466627.2481236. URL <http://doi.acm.org/10.1145/2466627.2481236>.

Oliver Bown, Benjamin Carey, and Arne Eigenfeldt. Manifesto for a Musebot Ensemble: A Platform for Live Interactive Performance Between Multiple Autonomous Musical Agents. In *Proceedings of the International Symposium of Electronic Art 2015 (ISEA 2015)*, 2015. URL http://isea2015.org/proceeding/submissions/ISEA2015_submission_141.pdf.

Liam Bray and Oliver Bown. Linear and non-linear composition systems: User experience in nodal and pro tools. In *Proceedings of the Australian Computer Music Association Conference*, 2014.

Liam Bray and Oliver Bown. Applying Core Interaction Design Principles to Computational Creativity. In *Proceedings of the Seventh International Conference on Computational Creativity*, 2016. URL <http://www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/Applying-Core-Interaction-Design-Principles-to-Computational-Creativity.pdf>.

Liam Bray, Oliver Bown, and Benjamin Carey. How Can We Deal With The Design Principle Of Visibility In Highly Encapsulated Computationally Creative Systems? In *Proceedings of the Eighth International Conference on Computational Creativity*, Atlanta, Georgia, USA, 2017.

Mason Bretan and Gil Weinberg. A survey of robotic musicianship. *Communications of the ACM*, 59(5):100–109, April 2016. ISSN 00010782. doi: 10.1145/2818994. URL <http://dl.acm.org/citation.cfm?doid=2930840.2818994>.

Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep Learning Techniques for Music Generation-A Survey. *arXiv preprint arXiv:1709.01620*, 2017.

Rodney A. Brooks. A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation*, 2(1):14–23, 1986. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1087032.

Rodney A. Brooks. Intelligence without reason. In *The artificial life route to artificial intelligence: Building embodied, situated agents*, pages 25–81. L. Erlbaum Associates Inc., NJ, USA, 1995. URL <http://idlebrain.yolasite.com/resources/Article%20-%20AI.pdf>.

Joanna Bryson. The Reactive Accompanist: Adaptation and Behavior Decomposition in a Music System. In Luc Steels, editor, *The Biology and Technology of Intelligent Autonomous Agents*, pages 365–376. Springer, Berlin, Heidelberg, 1995. ISBN 978-3-642-79631-9 978-3-642-79629-6. URL http://www.springerlink.com/index/10.1007/978-3-642-79629-6_15.

Bruce G Buchanan. Creativity at the Metalevel AAAI-2000 Presidential Address. *AI Magazine*, 22(3):16, 2001.

Antonio Camurri, Alessandro Catorcini, Carlo Innocenti, and Alberto Massari. Music and Multi-media Knowledge Representation and Reasoning: The HARP System. *Computer Music Journal*, 19(2):34, 1995. ISSN 01489267. doi: 10.2307/3680599. URL <http://www.jstor.org/stable/3680599?origin=crossref>.

Pietro Casella and Ana Paiva. Magenta: An architecture for real time automatic composition of background music. In *Intelligent Virtual Agents*, pages 224–232. Springer, 2001. URL http://link.springer.com/chapter/10.1007/3-540-44812-8_18.

Justine Cassell, Joseph Sullivan, Elizabeth Churchill, and Scott Prevost. *Embodied Conversational Agents*. MIT Press, 2000. ISBN 978-0-262-03278-0. Google-Books-ID: tHiKZGh9t7sC.

Joel Chadabe. *Electric sound: the past and promise of electronic music*. Prentice Hall, Upper Saddle River, N.J, 1997. ISBN 978-0-13-303231-4.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, December 2014. URL <http://arxiv.org/abs/1412.3555>. arXiv: 1412.3555.

Nick Collins. Drumtrack: Beat induction from an acoustic drum kit with synchronised scheduling. In *Proceedings of International Computer Music Conference (ICMC)*, 2005. URL <http://community.dur.ac.uk/nick.collins/research/drumtrack.pdf>.

Nick Collins. BBCut2: Integrating beat tracking and on-the-fly event analysis. *Journal of New Music Research*, 35(1):63–70, March 2006. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298210600696600. URL <http://www.tandfonline.com/doi/abs/10.1080/09298210600696600>.

Nick Collins. Reinforcement learning for live musical agents. In *Proceedings of the International Computer Music Conference (ICMC), Belfast*, 2008. URL <http://users.sussex.ac.uk/~nc81/research/rlforlivemusicalagents.pdf>.

Nick Collins. LL: Listening and learning in an interactive improvisation system. Technical report, University of Sussex, 2011.

Nick Collins. Towards Machine Musicians Who Have Listened to More Music Than Us: Audio Database-Led Algorithmic Criticism for Automatic Composition and Live Concert Systems. *Computers in Entertainment*, 14(3):1–14, January 2017. ISSN 15443574. doi: 10.1145/2967510. URL <http://dl.acm.org/citation.cfm?doid=3023312.2967510>.

Simon Colton. Creativity Versus the Perception of Creativity in Computational Systems. In *AAAI Spring Symposium: Creative Intelligent Systems*, volume 8, 2008.

Simon Colton and Geraint A. Wiggins. Computational Creativity: The Final Frontier? *Frontiers in Artificial Intelligence and Applications*, pages 21–26, 2012. ISSN 0922-6389. doi: 10.3233/978-1-61499-098-7-21. URL <http://www.medra.org/servlet/aliasResolver?alias=iospressISSNISBN&issn=0922-6389&volume=242&spage=21>.

Darrell Conklin. Multiple Viewpoint Systems for Music Classification. *Journal of New Music Research*, 42(1):19–26, March 2013. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2013.776611. URL <http://www.tandfonline.com/doi/abs/10.1080/09298215.2013.776611>.

Arshia Cont, Shlomo Dubnov, and Gérard Assayag. Anticipatory model of musical style imitation using collaborative and competitive reinforcement learning. In *Anticipatory behavior in adaptive learning systems*, pages 285–306. Springer, 2007. URL http://link.springer.com/chapter/10.1007/978-3-540-74262-3_16.

Mihaly Csikszentmihalyi. *Flow: The Psychology of Optimal Experience*. Harper Perennial Modern Classics, New York, 1 edition edition, July 2008. ISBN 978-0-06-133920-2.

Palle Dahlstedt and Mats G. Nordahl. Living melodies: Coevolution of sonic communication. *Leonardo*, 34(3):243–248, 2001. URL <http://www.mitpressjournals.org/doi/pdf/10.1162/002409401750287010>.

Roger B. Dannenberg. Style in Music. In Shlomo Argamon, Kevin Burns, and Shlomo Dubnov, editors, *The Structure of Style*, pages 45–57. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-12336-8 978-3-642-12337-5. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-642-12337-5_3. DOI: 10.1007/978-3-642-12337-5_3.

Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917, 2002. ISSN 00014966. doi: 10.1121/1.1458024. URL <http://scitation.aip.org/content/asa/journal/jasa/111/4/10.1121/1.1458024>.

Ken Dé guernel, Emmanuel Vincent, and Gérard Assayag. Probabilistic Factor Oracles for Multi-dimensional Machine Improvisation. *Computer Music Journal*, 42(2):52–66, 2018.

Miguel Delgado, Waldo Fajardo, and Miguel Molina-Solana. Inmmatusys: Intelligent Multiagent Music System. *Expert Systems with Applications*, 36(3):4574–4580, April 2009. ISSN 0957-4174. doi: 10.1016/j.eswa.2008.05.028. URL <http://dx.doi.org/10.1016/j.eswa.2008.05.028>.

Alexandre Donze, Rafael Valle, Ilge Akkaya, Sophie Libkind, Sanjit A Seshia, and David Wessel. Machine Improvisation with Formal Specifications. In *the Proceedings of the Joint Conference ICMC14-SMC14*, page 8, Athens, Greece, 2014.

Shlomo Dubnov and Gérard Assayag. Improvisation Planning and Jam Session Design using concepts of Sequence Variation and Flow Experience. In *Proceedings of Sound and Music Computing 2005*, page 7, Italy, 2005.

Shlomo Dubnov, Gérard Assayag, and Ran El-Yaniv. Universal Classification Applied to Musical Sequences. In *Proceedings of the 1998 International Computer Music Conference, ICMC 1998*, Ann Arbor, Michigan, USA, 1998.

Shlomo Dubnov, Stephen McAdams, and Roger Reynolds. Structural and affective aspects of music from statistical audio signal analysis. *Journal of the American Society for Information Science and Technology*, 57(11):1526–1536, September 2006. ISSN 15322882, 15322890. doi: 10.1002/asi.20429. URL <http://doi.wiley.com/10.1002/asi.20429>.

Shlomo Dubnov, Gerard Assayag, and Arshia Cont. Audio Oracle: A New Algorithm for Fast Learning of Audio Structures. In *Proceedings of International Computer Music Conference*, 2007. URL <https://hal.inria.fr/hal-00839072/document>.

Shlomo Dubnov, G. Assayag, and A. Cont. Audio Oracle Analysis of Musical Information Rate. In *Proceedings of the Fifth IEEE International Conference on Semantic Computing (ICSC)*, pages 567–571, September 2011. doi: 10.1109/ICSC.2011.106.

Douglas Eck and Jurgen Schmidhuber. A First Look at Music Composition using LSTM Recurrent Neural Networks. *Istituto Dalle Molle Di Studi Sull'Intelligenza Artificiale*, 103:11, 2002.

T. Eerola and J. K. Vuoskoski. A comparison of the discrete and dimensional models of emotion in music. *Psychology of Music*, 39(1):18–49, January 2011. ISSN 0305-7356, 1741-3087. doi: 10.1177/0305735610362821. URL <http://pom.sagepub.com/cgi/doi/10.1177/0305735610362821>.

Tuomas Eerola and Jonna K. Vuoskoski. A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli. *Music Perception: An Interdisciplinary Journal*, 30(3):307–340, February 2013. ISSN 07307829, 15338312. doi: 10.1525/mp.2012.30.3.307. URL <http://mp.ucpress.edu/cgi/doi/10.1525/mp.2012.30.3.307>.

Arne Eigenfeldt. Emergent Rhythms through Multi-agency in Max/MSP. In Richard Kronland-Martinet, Sølvi Ystad, and Kristoffer Jensen, editors, *Computer Music Modeling and Retrieval. Sense of Sounds*, number 4969 in Lecture Notes in Computer Science, pages 368–379. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-85034-2 978-3-540-85035-9. URL http://link.springer.com/chapter/10.1007/978-3-540-85035-9_26.

Arne Eigenfeldt. The Evolution of Evolutionary Software: Intelligent Rhythm Generation in Kinetic Engine. In Mario Giacobini, Anthony Brabazon, Stefano Cagnoni, Gianni A. Di Caro, Anikó Ekárt, Anna Isabel Esparcia-Alcázar, Muddassar Farooq, Andreas Fink, and Penousal Machado, editors, *Applications of Evolutionary Computing*, number 5484 in Lecture Notes in Computer Science, pages 498–507. Springer Berlin Heidelberg, 2009. ISBN 978-3-642-01128-3 978-3-642-01129-0. URL http://link.springer.com/chapter/10.1007/978-3-642-01129-0_56.

Arne Eigenfeldt. Coming together: Composition by negotiation. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 1433–1436. ACM, 2010. URL <http://dl.acm.org/citation.cfm?id=1874237>.

Arne Eigenfeldt. Multi-Agent Modeling of Complex Rhythmic Interactions in RealTime Performance. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.

Arne Eigenfeldt. Generating Structure—Towards Large-Scale Formal Generation. In *Proceedings of the Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. URL <http://musicalmetacreation.org/mume2014/content/proceedings/Generating%20Structure%20-%20Towards%20Large-scale%20Formal%20Generation.pdf>.

Arne Eigenfeldt. Exploring moment-form in generative music. In *Proceedings of 13th Sound and Music Conference*, Hamburg, Germany, 2016. ISBN 978-3-00-053700-4. URL https://www.researchgate.net/profile/Arne_Eigenfeldt/publication/306207035_EXPLORING_MOMENT-FORM_IN_GENERATIVE_MUSIC/links/57b35e5808aeac3177849721.pdf.

Arne Eigenfeldt. Musebots at One Year: A Review. In *Proceedings of the 4th International Workshop on Musical Metacreation (MUME 2016)*, 2016. ISBN 978-0-86491-397-5. URL https://www.researchgate.net/profile/Arne_Eigenfeldt/

publication/306206920_Musebots_at_One_Year_A_Review/links/57b35df608aeaf239baf1456.pdf.

Arne Eigenfeldt and Philippe Pasquier. A realtime generative music system using autonomous melody, harmony, and rhythm agents. In *Proceedings of the 12th Generative Art Conference*, 2009. URL <http://artscience-ebookshop.com/on/cic/GA2009Papers/p7.pdf>.

Arne Eigenfeldt and Philippe Pasquier. Real-Time Timbral Organisation: Selecting samples based upon similarity. *Organised Sound*, 15(02):159–166, August 2010. ISSN 1469-8153. doi: 10.1017/S1355771810000154. URL http://journals.cambridge.org/article_S1355771810000154.

Arne Eigenfeldt and Philippe Pasquier. Negotiated content: Generative soundscape composition by autonomous musical agents in Coming Together: Freesound. In *Proceedings of the Second International Conference on Computational Creativity, Mexico City*, pages 27–32, 2011. URL http://www.researchgate.net/profile/Arne_Eigenfeldt/publication/228411309_Negotiated_Content_Generative_Soundscape_Composition_by_Autonomous_Musical_Agents_in_Coming_Together_Freesound/links/0912f5093deae4b882000000.pdf.

Arne Eigenfeldt and Philippe Pasquier. A sonic eco-system of self-organising musical agents. In *9th European Event on Evolutionary and Biologically Inspired Music, Sound, Art and Design (EvoMusArt 2011)*, volume 6625, pages 283–292, Torino, 2011. Springer Verlag.

Arne Eigenfeldt and Philippe Pasquier. Creative Agents, Curatorial Agents, and Human-Agent Interaction in Coming Together. In *Proceedings of Sound and Music Computing 2012*, pages 181–186, Copenhagen, Denmark, 2012. URL <http://www.smcnetwork.org/system/files/smc2012-187.pdf>.

Arne Eigenfeldt, Oliver Bown, Philippe Pasquier, and Aengus Martin. Towards a Taxonomy of Musical Metacreation: Reflections on the First Musical Metacreation Weekend. In *Proceedings of the 2nd International Workshop on Musical Metacreation (MUME 2013)*, 2013. URL http://www.researchgate.net/profile/Arne_Eigenfeldt/publication/258258077_Towards_a_Taxonomy_of_Musical_Metacreation_Reflections_on_the_First_Musical_Metacreation_Weekend/links/02e7e527a2da4e5426000000.pdf.

Arne Eigenfeldt, Oliver Bown, and Benjamin Carey. Collaborative Composition with Creative Systems: Reflections on the First Musebot Ensemble. In *Proceedings of the Sixth International Conference on Computational Creativity June*, page 134, 2015. URL <http://axon.cs.byu.edu/ICCC2015proceedings/6.2Eigenfeldt.pdf>.

Aaron Einbond, Riccardo Borghesi, Diemo Schwarz, and Norbert Schnell. Introducing CatOracle: Corpus-based concatenative improvisation with the Audio Oracle algorithm. In *Proceedings of the International Computer Music Conference 2016*, pages 140–146, 2016.

Paul Ekman. An argument for basic emotions. *Cognition & Emotion*, 6(3):169–200, May 1992. ISSN 0269-9931. doi: 10.1080/02699939208411068. URL <http://www.informaworld.com/openurl?genre=article&doi=10.1080/02699939208411068&magic=crossref||D404A21C5BB053405B1A640AFFD44AE3>.

Mustafa Emirbayer and Ann Mische. What Is Agency? *American Journal of Sociology*, 103(4):962–1023, January 1998. ISSN 0002-9602, 1537-5390. doi: 10.1086/231294. URL <http://www.journals.uchicago.edu/doi/10.1086/231294>.

Jianyu Fan, Miles Thorogood, and Philippe Pasquier. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society*, 64(9):646–653, September 2016. ISSN 15494950. doi: 10.17743/jaes.2016.0044. URL <http://www.aes.org/e-lib/browse.cfm?elib=18373>.

Jianyu Fan, Kivanç Tatar, Miles Thorogood, and Philippe Pasquier. Ranking-Based Emotion Recognition for Experimental Music. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) 2017*, 2017.

Jacques Ferber, Olivier Gutknecht, and Fabien Michel. From agents to organizations: an organizational view of multi-agent systems. In *International Workshop on Agent-Oriented Software Engineering*, pages 214–230. Springer, 2003. URL http://link.springer.com/chapter/10.1007/978-3-540-24620-6_15.

Michael Filimowicz. *New directions in third wave human-computer interaction: volume 1 - technologies*. Springer Berlin Heidelberg, New York, NY, 2018. ISBN 978-3-319-73355-5.

Charles B. Fowler. The Museum of Music: A History of Mechanical Instruments. *Music Educators Journal*, 54(2):45, October 1967. ISSN 00274321. doi: 10.2307/3391092. URL <http://mej.sagepub.com/cgi/doi/10.2307/3391092>.

Alexandre R. J. François, E. Chew, and Dennis Thurmond. Performer-centered visual feedback for human-machine improvisation. *Computers in Entertainment*, 9(3):1–13, November 2011. ISSN 15443574. doi: 10.1145/2027456.2027459. URL <http://dl.acm.org/citation.cfm?doid=2027456.2027459>.

Alexandre RJ François, Isaac Schankler, and Elaine Chew. Mimi4x: An interactive audio-visual installation for high-level structural improvisation. *International Journal of Arts and Technology*, 6(2):138–151, 2013. URL <http://www.inderscienceonline.com/doi/abs/10.1504/IJART.2013.053557>.

Christopher Frayling. Research in Art and Design (Royal College of Art Research Papers, Vol 1, No 1, 1993/4). *Royal College of Art Research Papers*, 1(1), 1994. URL <http://researchonline.rca.ac.uk/384/>.

Ohad Fried, Zen Jin, and Reid Oda. AudioQuilt: 2d Arrangements of Audio Samples using Metric Learning and Kernelized Sorting. In *Proceedings of the International Conference on New Interfaces for Musical Expression*. Goldsmiths University of London, 2014. URL <http://www.nime2014.org/technical-programme/proceedings/>.

Liane Gabora. Cognitive mechanisms underlying the creative process. In *Proceedings of the 4th conference on Creativity & cognition*, pages 126–133. ACM, 2002. URL <http://dl.acm.org/citation.cfm?id=581730>.

Philip Galanter. What is generative art? Complexity theory as a context for art theory. In *Proceedings of the 6th Generative Art Conference*, 2003. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.90.2634>.

- R.A. Garcia. *Automatic generation of sound synthesis techniques*. PhD thesis, MIT, 2001.
- Dileep George. *How the brain might work: A hierarchical and temporal model for learning and recognition*. PhD thesis, Stanford University, 2008.
- Toby Gifford. Appropriate and Complementary Rhythmic Improvisation in an Interactive Music System. In Simon Holland, Katie Wilkie, Paul Mulholland, and Allan Seago, editors, *Music and Human-Computer Interaction*, Springer Series on Cultural Computing. Springer London, London, 2013. ISBN 978-1-4471-2989-9 978-1-4471-2990-5. URL <http://link.springer.com/10.1007/978-1-4471-2990-5>.
- Toby M. Gifford and Andrew R. Brown. Anticipatory timing in algorithmic rhythm generation. In *Proceedings of the Australasian Computer Music Conference 2010*, pages 21–28. Australasian Computer Music Association (ACMA), 2010. URL <http://eprints.qut.edu.au/33281/>.
- Marco Gillies, Bongshin Lee, Nicolas d’Alessandro, Joëlle Tilmanne, Todd Kulesza, Baptiste Caramiaux, Rebecca Fiebrink, Atau Tanaka, Jérémie Garcia, Frédéric Bevilacqua, Alexis Heloir, Fabrizio Nunnari, Wendy Mackay, and Saleema Amershi. Human-Centred Machine Learning. pages 3558–3565. ACM Press, 2016. ISBN 978-1-4503-4082-3. doi: 10.1145/2851581.2856492. URL <http://dl.acm.org/citation.cfm?doid=2851581.2856492>.
- Marcelo Gimenes and Eduardo Reck Miranda. An Ontomemetic Approach to Musical Intelligence. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.
- Marcelo Gimenes, Eduardo Reck Miranda, and Chris Johnson. Towards an intelligent rhythmic generator based on given examples: a memetic approach. In *Digital Music Research Network Summer Conference*, pages 41–46, 2005. URL http://cmr.soc.plymouth.ac.uk/publications/Gimenes_Glasgow_def.pdf.
- Marcelo Gimenes, Eduardo Reck Miranda, and Chris Johnson. Musicianship for robots with style. In *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, pages 197–202. ACM, 2007. URL <http://dl.acm.org/citation.cfm?id=1279778>.
- Rich Gold and John Maeda. *The Plenitude : Creativity, Innovation, and Making Stuff*. The MIT Press, Cambridge, US, 2007. ISBN 978-0-262-27399-2. URL <http://site.ebrary.com/lib/alltitles/docDetail.action?docID=10205843>.
- Antoni Gomila and Vincent C. Müller. Challenges for artificial cognitive systems. *Journal of Cognitive Science*, 13(4):453–469, 2012. URL <http://philpapers.org/rec/GOMCFA>.
- John M. Grey. Multidimensional perceptual scaling of musical timbres. *The Journal of the Acoustical Society of America*, 61(5):1270–1277, May 1977. ISSN 0001-4966. doi: 10.1121/1.381428. URL <http://scitation.aip.org.proxy.lib.sfu.ca/content/asa/journal/jasa/61/5/10.1121/1.381428>.
- Simon Harding, Jürgen Leitner, and Jürgen Schmidhuber. Cartesian Genetic Programming for Image Processing. In Rick Riolo, Ekaterina Vladislavleva, Marylyn D. Ritchie, and Jason H.

Moore, editors, *Genetic Programming Theory and Practice X*, Genetic and Evolutionary Computation, pages 31–44. Springer New York, 2013. ISBN 978-1-4614-6845-5 978-1-4614-6846-2. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-1-4614-6846-2_3. DOI: 10.1007/978-1-4614-6846-2_3.

Arno Hartholt, David Traum, Stacy C. Marsella, Ari Shapiro, Giota Stratou, Anton Leuski, Louis-Philippe Morency, and Jonathan Gratch. All Together Now: Introducing the Virtual Human Toolkit. In *13th International Conference on Intelligent Virtual Agents*, Edinburgh, UK, August 2013. URL <http://ict.usc.edu/pubs/All%20Together%20Now.pdf>.

Andrew Hawryshkewich, Philippe Pasquier, and Arne Eigenfeldt. Beatback: A Real-time Interactive Percussion System for Rhythmic Practise and Exploration. *Proceedings of the tenth International Conference on New Interfaces for Musical Expression*, pages 100–105, 2010. URL <http://www.nime.org/>.

Dorien Herremans, Ching-Hua Chuan, and Elaine Chew. A Functional Taxonomy of Music Generation Systems. *ACM Computing Surveys*, 50(5):1–30, September 2017. ISSN 03600300. doi: 10.1145/3108242. URL <http://dl.acm.org/citation.cfm?doid=3145473.3108242>.

Francis Heylighen. Stigmergy as a universal coordination mechanism I: Definition and components. *Cognitive Systems Research*, 38:4–13, June 2016. ISSN 13890417. doi: 10.1016/j.cogsys.2015.12.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S1389041715000327>.

William Hsu. Strategies for Managing Timbre and Interaction in Automatic Improvisation Systems. *Leonardo Music Journal*, 20(1):33–39, 2010. ISSN 1531-4812. URL <https://muse-jhu-edu.proxy.lib.sfu.ca/article/404089>.

David Brian Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, Cambridge, UNITED STATES, 2014. ISBN 978-0-262-27596-5. URL <http://ebookcentral.proquest.com/lib/sfu-ebooks/detail.action?docID=3338552>.

International Telecommunication Union. Recommendation ITU-R BS.468: Measurement of audio-frequency noise voltage level in sound broadcasting. In *ITU-R recommendations: BS series; Broadcasting service (sound)*. International Telecommunication Union, Geneva, 1990.

Anna Jordanous. A Standardised Procedure for Evaluating Creative Systems: Computational Creativity Evaluation Based on What it is to be Creative. *Cognitive Computation*, 4(3):246–279, September 2012. ISSN 1866-9956, 1866-9964. doi: 10.1007/s12559-012-9156-1. URL <http://link.springer.com/10.1007/s12559-012-9156-1>.

Eric Kandel. *Reductionism in Art and Brain Science: Bridging the Two Cultures*. Columbia University Press, New York, 1 edition edition, August 2016. ISBN 978-0-231-17962-1.

Ismo Kauppinen. Methods for detecting impulsive noise in speech and audio signals. In *Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*, volume 2, pages 967–970. IEEE, 2002. URL <http://ieeexplore.ieee.org/abstract/document/1028251/>.

John P. Kimball. *Syntax and Semantics*. Academic Press, 1975. ISBN 978-0-12-785423-6.

Alexis Kirke and Eduardo Miranda. A Biophysically Constrained Multi-Agent Systems Approach to Algorithmic Composition with Expressive Performance. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.

Alexis Kirke and Eduardo Miranda. A Multi-Agent Emotional Society Whose Melodies Represent its Emergent Social Hierarchy and Are Generated by Agent Communications. *Journal of Artificial Societies and Social Simulation*, 18(2):16, 2015. ISSN 1460-7425. doi: 10.18564/jasss.2679. URL <http://jasss.soc.surrey.ac.uk/18/2/16.html>.

Teuvo Kohonen. Self-organized formation of topologically correct feature maps. *Biological cybernetics*, 43(1):59–69, 1982. URL <http://link.springer.com/article/10.1007/BF00337288>.

Teuvo Kohonen. The self-organizing map. *Neurocomputing*, 21(1–3):1–6, November 1998. ISSN 0925-2312. doi: 10.1016/S0925-2312(98)00030-7. URL <http://www.sciencedirect.com/science/article/pii/S0925231298000307>.

Olivier Lartillot, Donato Cereghetti, Kim Eliard, and Didier Grandjean. A simple, high-yield method for assessing structural novelty. In *Proceedings of the 3rd International Conference on Music & Emotion (ICME3), Jyväskylä, Finland, 11th-15th June 2013. Geoff Luck & Olivier Brabant (Eds.). ISBN 978-951-39-5250-1*. University of Jyväskylä, Department of Music, 2013. URL <https://jyx.jyu.fi/dspace/handle/123456789/41611>.

Arnaud Lefebvre and Thierry Lecroq. A Heuristic For Computing Repeats With A Factor Oracle: Application To Biological Sequences. *International Journal of Computer Mathematics*, 79(12):1303–1315, January 2002. ISSN 0020-7160, 1029-0265. doi: 10.1080/00207160214653. URL <http://www.tandfonline.com/doi/abs/10.1080/00207160214653>.

Aaron Levisohn and Philippe Pasquier. BeatBender: subsumption architecture for autonomous rhythm generation. In *Proceedings of the ACM International Conference on Advances in Computer Entertainment Technologies (ACE 2008)*, pages 51–58, Yokohama, Japan, 2008. URL <http://eprints.iat.sfu.ca/883/>.

Benjamin Lévy, Georges Bloch, and Gérard Assayag. OMaxist dialectics. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, pages 137–140, 2012. URL <https://hal.archives-ouvertes.fr/hal-00706662/>.

George E. Lewis. Too many notes: Computers, complexity and culture in voyager. *Leonardo Music Journal*, 10:33–39, 2000. URL <http://www.mitpressjournals.org/doi/abs/10.1162/096112100570585>.

Adam Linson, Chris Dobbyn, George E. Lewis, and Robin Laney. A Subsumption Agent for Collaborative Free Improvisation. *Computer Music Journal*, 39(4):96–115, 2015. ISSN 1531-5169. URL https://muse-jhu-edu.proxy.lib.sfu.ca/journals/computer_music_journal/v039/39.4.linson.html.

Zachary C. Lipton, John Berkowitz, and Charles Elkan. A Critical Review of Recurrent Neural Networks for Sequence Learning. *arXiv:1506.00019 [cs]*, May 2015. URL <http://arxiv.org/abs/1506.00019>. arXiv: 1506.00019.

Steven R. Livingstone, Ralf Mühlberger, Andrew R. Brown, and Andrew Loch. Controlling musical emotionality: an affective computational architecture for influencing musical emotions. *Digital Creativity*, 18(1):43–53, March 2007. ISSN 1462-6268, 1744-3806. doi: 10.1080/14626260701253606. URL <http://www.tandfonline.com/doi/abs/10.1080/14626260701253606>.

Russolo Luigi. *The Art of Noise*. A Great Bear Pamphlet, 1967.

Michael F. Lynch. Motivation, Microdrives and Microgoals in Mockingbird. In *Proceedings of 3rd International Workshop on Musical Metacreation (MUME 2014)*, North Carolina, USA, 2014. URL <http://musicalmetacreation.org/mume2014/content/proceedings/Motivation,%20Microdrives%20and%20Microgoals%20in%20Mockingbird.pdf>.

M. Macret and P. Pasquier. Automatic Design of Sound Synthesizers As Pure Data Patches Using Coevolutionary Mixed-typed Cartesian Genetic Programming. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation*, GECCO '14, pages 309–316, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2662-9.

Ethan Manilow, Prem Seetharaman, and Bryan Pardo. The northwestern university source separation library. Proceedings of the 19th International Society of Music Information Retrieval Conference (ISMIR 2018), Paris, France, September 23-27, 2018.

Ethan Manilow, Prem Seetharaman, Fatemeh Pishdadian, and Bryan Pardo. NUSSL: the northwestern university source separation library. <https://github.com/interactiveaudiolab/nussl>, 2018.

Aengus Martin and Oliver Bown. The Agent Designer Toolkit. In *Proceedings of the 9th ACM Conference on Creativity & Cognition*, C&C '13, pages 386–387, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2150-1. doi: 10.1145/2466627.2481211. URL <http://doi.acm.org/10.1145/2466627.2481211>.

Aengus Martin, C. T. Jin, André van Schaik, and William L. Martens. Partially observable Markov decision processes for interactive music systems. In *Proceedings of the International Computer Music Conference*, 2010. URL <http://www.ee.usyd.edu.au/carlab/CARlabPublicationsData/PDF/2010%20Martin%20In%20International%20Computer%20Music%20Conference-3237008059/2010%20Martin%20In%20International%20Computer%20Music%20Conference.pdf>.

Aengus Martin, Craig T. Jin, and Oliver Bown. A Toolkit for Designing Interactive Musical Agents. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, OzCHI '11, pages 194–197, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-1090-1. doi: 10.1145/2071536.2071567. URL <http://doi.acm.org/10.1145/2071536.2071567>.

Aengus Martin, Craig T. Jin, and Oliver Bown. Implementation of a real-time musical decision-maker. In *Proceedings of the Australasian Computer Music Conference*, 2012. URL <http://www.maths.tcd.ie/~hobo/papers/2012acmc.pdf>.

Aengus Martin, Craig T. Jin, Ben Carey, and Oliver Bown. Creative experiments using a system for learning high-level performance structure in ableton live. In *Proceedings of the Sound and Music Computing Conference*, 2012. URL <http://smcnetwork.org/system/files/smc2012-206.pdf>.

Joao M. Martins and Eduardo R. Miranda. A connectionist architecture for the evolution of rhythms. In *Applications of Evolutionary Computing*, pages 696–706. Springer, 2006. URL http://link.springer.com/chapter/10.1007/11732242_66.

Joao M. Martins and Eduardo R. Miranda. Emergent rhythmic phrases in an A-Life environment. In *Proceedings of ECAL 2007 Workshop on Music and Artificial Life (MusicAL 2007)*, pages 10–14, 2007. URL http://cmr.soc.plymouth.ac.uk/publications/MusicAL_Martins.pdf.

Joao M. Martins and Eduardo R. Miranda. Breeding rhythms with artificial life. In *Proceedings of the Sound and Music Conference*. Citeseer, 2008. URL <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.178.3253&rep=rep1&type=pdf>.

James B. Maxwell, Philippe Pasquier, and Eigenfeldt Eigenfeldt. Hierarchical Sequential Memory for Music: A Cognitive Model. In *Proceedings of the 10th International Conference for Music Information Retrieval*, 2009.

James B. Maxwell, Arne Eigenfeldt, Philippe Pasquier, and N. Gonzalez Thomas. MusiCOG: A cognitive architecture for music learning and generation. In *Proceedings of the Sound and Music Computing Conference*, page 9, 2012. URL <http://smcnetwork.org/system/files/smc2012-255.pdf>.

Jon McCormack and Oliver Bown. Life's what you make: Niche construction and evolutionary art. In *Workshops on Applications of Evolutionary Computation*, pages 528–537. Springer, 2009. URL http://link.springer.com/chapter/10.1007/978-3-642-01129-0_59.

Jon McCormack, Peter McIlwain, Aidan Lane, and Alan Dorin. Generative composition with Nodal. In *Workshop on music and artificial life (part of ECAL 2007), Lisbon, Portugal, 2007*. URL http://www.researchgate.net/profile/Aidan_Lane/publication/228749312_Generative_composition_with_Nodal/links/004635289f6dc0d7d600000.pdf.

Julian F. Miller, editor. *Cartesian Genetic Programming*. Natural Computing Series. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-17309-7 978-3-642-17310-3. URL <http://link.springer.com/10.1007/978-3-642-17310-3>.

Marvin Minsky. *The society of mind*. Simon and Schuster, New York, N.Y, 1986. ISBN 978-0-671-60740-1.

Eduardo R. Miranda, Alexis Kirke, and Qijun Zhang. Artificial Evolution of Expressive Performance of Music: An Imitative Multi-Agent Systems Approach. *Computer Music Journal*, 34(1): 80–96, 2010. ISSN 0148-9267. URL <http://www.jstor.org.proxy.lib.sfu.ca/stable/25653532>.

Eduardo Reck Miranda and Al Biles, editors. *Evolutionary computer music*. Springer, London, 2007. ISBN 978-1-84628-599-8. OCLC: ocm80332658.

Tom Michael Mitchell. *Machine Learning*. McGraw-Hill Education, March 1997. ISBN 978-0-07-042807-2.

Brian C. J. Moore, Brian R. Glasberg, and Thomas Baer. A Model for the Prediction of Thresholds, Loudness, and Partial Loudness. *Journal of Audio Engineering Society*, 45(4):224–240, 1997. URL <http://www.aes.org/e-lib/browse.cfm?elib=10272>.

Julian Moreira, Pierre Roy, and François Pachet. Virtualband: Interacting with Stylistically Consistent Agents. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 341–346, Brazil, 2013. URL http://ismir2013.ismir.net/wp-content/uploads/2013/09/277_Paper.pdf.

Meinard Müller. *Fundamentals of Music Processing*. Springer International Publishing, Cham, 2015. ISBN 978-3-319-21944-8 978-3-319-21945-5. URL <http://link.springer.com/10.1007/978-3-319-21945-5>.

D. Murray-Rust, A. Smaill, and M.C. Maya. VirtuaLatin - towards a musical multi-agent system. In *Sixth International Conference on Computational Intelligence and Multimedia Applications, 2005*, pages 17–22, August 2005. doi: 10.1109/ICCIMA.2005.59.

Dave Murray-Rust and Alan Smaill. Towards a model of musical interaction and communication. *Artificial Intelligence*, 175(9-10):1697–1721, June 2011. ISSN 00043702. doi: 10.1016/j.artint.2011.01.002. URL <http://linkinghub.elsevier.com/retrieve/pii/S0004370211000038>.

David Murray-Rust. *Musical Acts and Musical Agents: theory, implementation and practice*. PhD thesis, 2008. URL <https://www.era.lib.ed.ac.uk/handle/1842/2561>.

David Murray-Rust, Alan Smaill, and Michael Edwards. MAMA: An Architecture for Interactive Musical Agents. *Frontiers in Artificial Intelligence and Applications*, 141:36, 2006.

DS Murray-Rust and Alan Smaill. Musical acts and musical agents. *Proceedings of the 5th Open Workshop of MUSICNETWORK: Integration of Music in Multimedia Applications (to Appear)*, 10, 2005.

Maria Navarro, Juan Manuel Corchado, and Yves Demazeau. A Musical Composition Application Based on a Multiagent System to Assist Novel Composers. *International Conference on Computational Creativity*, 2014.

Maria Navarro, Juan Manuel Corchado, and Yves Demazeau. MUSIC-MAS: Modeling a harmonic composition system with virtual organizations to assist novice composers. *Expert Systems with Applications*, 57:345–355, September 2016. ISSN 09574174. doi: 10.1016/j.eswa.2016.01.058. URL <http://linkinghub.elsevier.com/retrieve/pii/S0957417416300227>.

Jérôme Nika and Marc Chemillier. Improtek: integrating harmonic controls into improvisation in the filiation of OMax. In *International Computer Music Conference (ICMC)*, pages 180–187, 2012. URL <https://hal.archives-ouvertes.fr/hal-01059330/>.

Jérôme Nika, José Echeveste, Marc Chemillier, and Jean-Louis Giavitto. Planning Human-Computer Improvisation. In *International Computer Music Conference*, page 330, 2014. URL <https://hal.archives-ouvertes.fr/hal-01053834/>.

Jérôme Nika, Dimitri Bouche, Jean Bresson, Marc Chemillier, and Gérard Assayag. Guided improvisation as dynamic calls to an offline model. In *Sound and Music Computing (SMC)*, Maynooth, Ireland, July 2015. URL <https://hal.archives-ouvertes.fr/hal-01184642>.

Jérôme Nika, Marc Chemillier, and Gérard Assayag. ImprotoK: Introducing Scenarios into Human-Computer Music Improvisation. *Computers in Entertainment*, 14(2):1–27, January 2017. ISSN 15443574. doi: 10.1145/3022635. URL <http://dl.acm.org/citation.cfm?doid=3023311.3022635>.

Doug Van Nort, Pauline Oliveros, and Jonas Braasch. Electro/Acoustic Improvisation and Deeply Listening Machines. *Journal of New Music Research*, 42(4):303–324, December 2013. ISSN 0929-8215. doi: 10.1080/09298215.2013.860465. URL <http://dx.doi.org/10.1080/09298215.2013.860465>.

Andrew Ortony and Terence J. Turner. What's basic about basic emotions? *Psychological review*, 97(3):315, 1990. URL <http://psycnet.apa.org/journals/rev/97/3/315/>.

François Pachet. Rhythms as emerging structures. In *Proceedings of 2000 International Computer Music Conference, Berlin, ICMA*, 2000. URL http://www.researchgate.net/profile/Francois_Pachet/publication/243602024_Rhythms_as_emerging_structures/links/0deec52909fbb73f05000000.pdf.

François Pachet. Beyond the cybernetic jam fantasy: The continuator. *Computer Graphics and Applications, IEEE*, 24(1):31–35, 2004. URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1255806.

François Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003. URL <http://www.tandfonline.com/doi/abs/10.1076/jnmr.32.3.333.16861>.

Ana Paiva, Gerd Andersson, Kristina Höök, Dário Mourão, Marco Costa, and Carlos Martinho. Sentoy in fantasy: Designing an affective sympathetic interface to a computer game. *Personal and Ubiquitous Computing*, 6(5-6):378–389, 2002. URL <http://dl.acm.org/citation.cfm?id=592611>.

Jaak Panksepp. *Affective Neuroscience: The Foundations of Human and Animal Emotions*. Oxford University Press, September 1998. ISBN 978-0-19-802567-2.

Philippe Pasquier, Arne Eigenfeldt, Oliver Bown, and Shlomo Dubnov. An Introduction to Musical Metacreation. *Computers in Entertainment*, 14(2):1–14, January 2017. ISSN 15443574. doi: 10.1145/2930672. URL <http://dl.acm.org/citation.cfm?doid=3023311.2930672>.

Alison Pease and Simon Colton. On impact and evaluation in computational creativity: A discussion of the Turing test and an alternative proposal. In *Proceedings of the AISB symposium on AI and Philosophy*, 2011.

G. Peeters. A large set of audio features for sound description (similarity and classification) in the CUIDADO project. Technical report, IRCAM, 2004.

Geoffroy Peeters, Bruno L. Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. The timbre toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 2011.

M. Peter. Milieus of creativity: The role of places, environments, and spatial. In Peter Meusburger, Joachim Funke, and Edgar Wunder, editors, *Milieus of creativity: an interdisciplinary approach to spatiality of creativity*, number v. 2 in Knowledge and space, pages 97–153. Springer, Dordrecht : [Heidelberg], 2009. ISBN 978-1-4020-9876-5.

David Plans and Davide Morelli. Using Coevolution in Music Improvisation. In *A-life for Music: Music and Computer Models of Living Systems*. A-R Editions, Inc., 2011. ISBN 978-0-89579-673-8.

Reinier Plomp and Willem Johannes Maria Levelt. Tonal consonance and critical bandwidth. *The journal of the Acoustical Society of America*, 38(4):548–560, 1965. URL <http://asa.scitation.org/doi/abs/10.1121/1.1909741>.

H. F. Pollard and E. V. Jansson. A Tristimulus Method for the Specification of Musical Timbre. *Acta Acustica united with Acustica*, 51(3):162–171, August 1982.

Mirjana Prpa, Kivanç Tatar, Bernhard E. Riecke, and Philippe Pasquier. The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality. In *Virtual, Augmented and Mixed Reality, 9th International Conference, VAMR 2017, Held as Part of HCI International 2017, Proceedings*, Vancouver, 2017. Springer. ISBN 978-3-319-57986-3. URL <http://www.springer.com/gp/book/9783319579863>.

Martin L. Puterman. *Markov decision processes: discrete stochastic dynamic programming*. Wiley series in probability and mathematical statistics. Applied probability and statistics section. John Wiley & Sons, New York, 1994. ISBN 978-0-471-61977-2.

A. Rauber, D. Merkl, and M. Dittenbach. The growing hierarchical self-organizing map: exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks*, 13(6):1331–1341, November 2002. ISSN 1045-9227. doi: 10.1109/TNN.2002.804221.

Jaume Rigau, Miquel Feixas, and Mateu Sbert. Informational aesthetics measures. *IEEE Computer Graphics and Applications*, 28(2):24–34, 2008. URL https://www.researchgate.net/profile/Mateu_Sbert/publication/5501365_Informational_Aesthetics_Measures/links/0912f51086574986ad000000.pdf.

Graeme Ritchie. Evaluating Quality in Creative Systems, 2014. URL http://videolectures.net/ascc2013_ritchie_systems/.

Curtis Roads. *Microsound*. The MIT Press, Cambridge, Mass., August 2004. ISBN 9780262681544.

Curtis Roads. *Composing electronic music: a new aesthetic*. Oxford University Press, Oxford, 2015. ISBN 978-0-19-537324-0.

Dana Ron, Yoram Singer, and Naftali Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine learning*, 25(2-3):117–149, 1996. URL <http://link.springer.com/article/10.1023/A:1026490906255>.

Robert Rowe. Machine Listening and Composing with Cypher. *Computer Music Journal*, 16(1):43, 1992. ISSN 01489267. doi: 10.2307/3680494. URL <http://www.jstor.org/stable/3680494?origin=crossref>.

James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39(6):1161–1178, December 1980. ISSN 0022-3514. doi: 10.1037/h0077714. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=1981-25062-001&site=ehost-live>.

Stuart J. (Stuart Jonathan) Russell and Peter Norvig. *Artificial intelligence: a modern approach*. Prentice Hall series in artificial intelligence. Prentice Hall, Upper Saddle River, N.J, 3rd edition, 2010. ISBN 978-0-13-207148-2.

Pablo Azevedo Sampaio, Geber Ramalho, and Patrícia Tedesco. CinBalada: a multiagent rhythm factory. *Journal of the Brazilian Computer Society*, 14(3):31–49, 2008. URL http://www.scielo.br/scielo.php?pid=S0104-65002008000300004&script=sci_arttext&tlang=es.

Norbert Schnell, Axel Röbel, Diemo Schwarz, Geoffroy Peeters, and Riccardo Borghesi. MuBu and friends—assembling tools for content based real-time interactive audio processing in Max/MSP. In *Proceedings of International Computer Music Conference (ICMC)*, 2009.

Martin Siefkes. Style: A new semiotic view on an old problem. *Kodikas/Code. Ars Semeiotica*, 34(1-2), 2011.

Herbert A. Simon. *The new science of management decision*, volume xii of *The Ford distinguished lectures*. Harper & Brothers, New York, NY, US, 1960. DOI: 10.1037/13978-000.

S. N. Sivanandam and S. N. Deepa. *Introduction to genetic algorithms*. Springer, Berlin ; New York, 2007. ISBN 978-3-540-73189-4.

Denis Smalley. Spectromorphology: explaining sound-shapes. *Organised Sound*, 2(02):107–126, August 1997. ISSN 1469-8153. doi: 10.1017/S1355771897009059. URL http://journals.cambridge.org/article_S1355771897009059.

Benjamin D. Smith and W. Scott Deal. ML.* Machine Learning Library as a Musical Partner in the Computer-Acoustic Composition Flight. In *the Proceedings of the Joint Conference ICMC14-SMC14*, volume 2014, Athens, Greece, 2014. URL <http://www.smc-conference.net/smc-icmc-2014/images/proceedings/OS19-B09-ML.pdf>.

Benjamin D. Smith and Guy E. Garnett. Reinforcement Learning and the Creative, Automated Music Improviser. In Penousal Machado, Juan Romero, and Adrian Carballal, editors, *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, number 7247 in Lecture Notes in Computer Science, pages 223–234. Springer Berlin Heidelberg, April 2012. ISBN 978-3-642-29141-8 978-3-642-29142-5. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-642-29142-5_20.

Benjamin D. Smith and Guy E. Garnett. Unsupervised Play: Machine Learning Toolkit for Max. In *the Proceedings of International Conference on New Interfaces for Musical Expression 2012*, 2012. URL http://www.nime.org/proceedings/2012/nime2012_68.pdf.

Karlheinz Stockhausen. Four Criteria of Electronic Music with Examples from Kontakte, 1972. URL <https://www.youtube.com/watch?v=7xyGtI7KKIY&list=PLRBdTyZ761vAF0tZvocPjpRVTL6htJzoP>.

Bob L. Sturm, Oded Ben-Tal, Úna Monaghan, Nick Collins, Dorien Herremans, Elaine Chew, Gaëtan Hadjeres, Emmanuel Deruty, and François Pachet. Machine learning research that matters for music creation: A case study. *Journal of New Music Research*, 48(1):36–55, January 2019. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1515233. URL <https://www.tandfonline.com/doi/full/10.1080/09298215.2018.1515233>.

David J.T Sumpter and Madeleine Beekman. From nonlinearity to optimality: pheromone trail foraging by ants. *Animal Behaviour*, 66(2):273–280, August 2003. ISSN 00033472. doi: 10.1006/anbe.2003.2224. URL <http://linkinghub.elsevier.com/retrieve/pii/S000334720392224X>.

Greg Surges and Shlomo Dubnov. Feature selection and composition using PyOracle. In *Ninth Artificial Intelligence and Interactive Digital Entertainment Conference*, 2013. URL http://www.pucktronix.com/media/papers/Surges_Dubnov_MuME2013_final.pdf.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning : An Introduction*. Adaptive Computation and Machine Learning. A Bradford Book, Cambridge, Mass, 1998. ISBN 978-0-262-19398-6. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=nlebk&AN=1094&site=ehost-live>.

T. Takala, J. Hahn, L. Gritz, J. Geigel, and J. Lee. Using physically based models and genetic algorithms for functional composition of sound signals, synchronized to animated motion. In *Proceedings of the International Computer Music Conference*, pages 180–185, 1993.

Kıvanç Tatar and Philippe Pasquier. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, June 2017. ISBN 978-1-77287-019-0.

Kıvanç Tatar and Philippe Pasquier. Musical agents: A typology and state of the art towards Musical Metacreation. *Journal of New Music Research*, 48(1):1–50, September 2018. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2018.1511736.

Kıvanç Tatar, Matthieu Macret, and Philippe Pasquier. Automatic Synthesizer Preset Generation with PresetGen. *Journal of New Music Research*, 45(2):124–144, April 2016. ISSN 0929-8215. doi: 10.1080/09298215.2016.1175481. URL <http://dx.doi.org/10.1080/09298215.2016.1175481>.

Kıvanç Tatar, Philippe Pasquier, and Remy Siu. REVIVE: An Audio-visual Performance with Musical and Visual AI Agents. pages 1–6. ACM Press, 2018. ISBN 978-1-4503-5621-3. doi: 10.1145/3170427.3177771. URL <http://dl.acm.org/citation.cfm?doid=3170427.3177771>.

Kıvanç Tatar, Philippe Pasquier, and Remy Siu. Audio-based Musical Artificial Intelligence and Audio-Reactive Visual Agents in Revive. New York, NY, US, 2019. International Computer Music Association.

Belinda Thom. BoB: An Interactive Improvisational Music Companion. In *Proceedings of the Fourth International Conference on Autonomous Agents*, AGENTS '00, pages 309–316, New York, NY, USA, 2000. ACM. ISBN 1-58113-230-1. doi: 10.1145/336595.337510. URL <http://doi.acm.org/10.1145/336595.337510>.

Belinda Thom. Unsupervised learning and interactive jazz/blues improvisation. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*, pages 652–657, 2000. URL <http://www.aaai.org/Papers/AAAI/2000/AAAI00-100.pdf>.

Belinda Thom. Interactive improvisational music companionship: A user-modeling approach. *User Modeling and User-Adapted Interaction*, 13(1-2):133–177, 2003. URL <http://link.springer.com/article/10.1023/A:1024014923940>.

Nicolas Gonzalez Thomas, Philippe Pasquier, Arne Eigenfeldt, and James B. Maxwell. A Methodology for the Comparison of Melodic Generation Models Using Meta-Melo. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 561–566, Brazil, 2013. ISBN 978-0-615-90065-0. URL http://ismir2013.ismir.net/wp-content/uploads/2013/09/228_Paper.pdf.

Kristinn Thórisson and Helgi Helgasson. Cognitive Architectures and Autonomy: A Comparative Review. *Journal of Artificial General Intelligence*, 3(2):1–30, January 2012. ISSN 1946-0163. doi: 10.2478/v10229-011-0015-3. URL <http://www.degruyter.com/view/j/jagi.2012.3.issue-2/v10229-011-0015-3/v10229-011-0015-3.xml>.

Peter M Todd and Gregory M Werner. Frankensteinian methods for evolutionary music. In *Musical networks: parallel distributed perception and performance*, pages 313–340. MIT Press/Bradford Books, Cambridge, MA, 1999.

Petri Toivainen, Mauri Kaipainen, and Jukka Louhivuori. Musical timbre: Similarity ratings correlate with computational feature space distances*. *Journal of New Music Research*, 24(3):282–298, September 1995. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298219508570686. URL <http://www.tandfonline.com/doi/abs/10.1080/09298219508570686>.

Barry Truax. *Acoustic Communication*. Greenwood Publishing Group, 2001. ISBN 978-1-56750-536-8.

Alan M. Turing. Computing machinery and intelligence. *Mind*, 59(236):433–460, 1950.

Kai Tuuri and Tuomas Eerola. Formulating a Revised Taxonomy for Modes of Listening. *Journal of New Music Research*, 41(2):137–152, June 2012. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298215.2011.614951. URL <http://www.tandfonline.com/doi/abs/10.1080/09298215.2011.614951>.

Leo Kazuhiro Ueda and Fabio Kon. Andante: A mobile musical agents infrastructure. In *Proceedings of the 9th Brazilian Symposium on Computer Music*, pages 87–94, 2003. URL http://gsd.ime.usp.br/sbcm/2003/papers/rLeo_Ueda.pdf.

Rafael Valle, Alexandre Donzé, Daniel J. Fremont, Ilge Akkaya, Sanjit A. Seshia, Adrian Freed, and David Wessel. Specification Mining for Machine Improvisation with Formal Specifications. *Computers in Entertainment*, 14(3):1–20, January 2017. ISSN 15443574. doi: 10.1145/2967504. URL <http://dl.acm.org/citation.cfm?doid=3023312.2967504>.

Edgard Varese and Chou Wen-chung. The liberation of Sound. *Perspectives of New Music*, 5(1):11–19, 1966. URL https://www.jstor.org/stable/832385?origin=JSTOR-pdf&seq=1#page_scan_tab_contents.

Rosa Maria Vicari, Lauro Nakayama, Rodolfo Daniel Wulffhorst, Leandro Lesqueves Costalonga, and Evandro Manara Miletto. The Musical Interactions within Community Agents. *Agent-Based Simulation Conference*, 2005.

J. N. Vold. *A study of musical problem solving behavior in kindergarten children and a comparison with other aspects of creative behavior*. PhD thesis, University of Alabama, 1986.

Marcelo M. Wanderley. Quantitative Analysis of Non-obvious Performer Gestures. In G. Goos, J. Hartmanis, J. van Leeuwen, Ipke Wachsmuth, and Timo Sowa, editors, *Gesture and Sign Language in Human-Computer Interaction*, volume 2298, pages 241–253. Springer Berlin Heidelberg, Berlin, Heidelberg, 2002. ISBN 978-3-540-43678-2 978-3-540-47873-7. doi: 10.1007/3-540-47873-6_26. URL http://link.springer.com/10.1007/3-540-47873-6_26.

Marcelo M. Wanderley, Bradley W. Vines, Neil Middleton, Cory McKay, and Wesley Hatch. The Musical Significance of Clarinetists’ Ancillary Gestures: An Exploration of the Field. *Journal of New Music Research*, 34(1):97–113, March 2005. ISSN 0929-8215, 1744-5027. doi: 10.1080/09298210500124208. URL <http://www.tandfonline.com/doi/abs/10.1080/09298210500124208>.

Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014. URL http://acsweb.ucsd.edu/~chw160/pdf/mume2014_vmo_Wang_Dubnov.pdf.

Cheng-i Wang and Shlomo Dubnov. Context-Aware Hidden Markov Models of Jazz Music with Variable Markov Oracle. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*, Atlanta, Georgia, USA, 2017.

Cheng-I Wang, Jennifer Hsu, and Shlomo Dubnov. Machine Improvisation with Variable Markov Oracle: Toward Guided and Structured Improvisation. *Computers in Entertainment*, 14(3):1–18, January 2017. ISSN 15443574. doi: 10.1145/2905371. URL <http://dl.acm.org/citation.cfm?doid=3023312.2905371>.

Peter R. Webster. Conceptual Bases for Creative Thinking in Music. In *Music and Child Development*, pages 158–174. Springer, New York, NY, 1987. ISBN 978-1-4613-8700-8 978-1-4613-8698-8. URL https://link-springer-com.proxy.lib.sfu.ca/chapter/10.1007/978-1-4613-8698-8_8. DOI: 10.1007/978-1-4613-8698-8_8.

Kaare Wehn. Using Ideas from Natural Selection to Evolve Synthesized Sounds. In *Proceedings of the Digital Audio Effects DAFX98 workshop*, pages 159–167, Barcelona, 1998.

Gerhard Weiss. *Multiagent systems*. Intelligent robotics and autonomous agents. The MIT Press, Cambridge, Massachusetts, second edition. edition, 2013. ISBN 978-0-262-01889-0.

Ian Whalley. PIWeCS: enhancing human/machine agency in an interactive composition system. *Organised Sound*, 9(02), August 2004. ISSN 1355-7718, 1469-8153. doi: 10.1017/S135577180400024X. URL http://www.journals.cambridge.org/abstract_S135577180400024X.

Mitchell Whitelaw. *Metacreation: art and artificial life*. MIT Press, Cambridge, Mass, 2004. ISBN 9780262232340.

Geraint A. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7):449–458, November 2006. ISSN 09507051. doi: 10.1016/j.knosys.2006.04.009. URL <http://linkinghub.elsevier.com/retrieve/pii/S0950705106000645>.

Geraint A. Wiggins. Searching for computational creativity. *New Generation Computing*, 24(3):209–222, September 2006. ISSN 0288-3635, 1882-7055. doi: 10.1007/BF03037332. URL <http://link.springer.com/10.1007/BF03037332>.

Michael Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, June 2009. ISBN 9780470519462.

Rodolfo Daniel Wulffhorst, Lauro Nakayama, and Rosa Maria Vicari. A Multiagent Approach for Musical Interactive Systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '03, pages 584–591, New York, NY, USA, 2003. ACM. ISBN 1-58113-683-8. doi: 10.1145/860575.860669. URL <http://doi.acm.org/10.1145/860575.860669>.

Matthew Yee-King and Mark d’Inverno. Experience driven design of creative systems. In *Title: Proceedings of the 7th Computational Creativity Conference (ICCC 2016)*. Universite Pierre et Marie Curie, 2016. URL <http://www.computationalcreativity.net/iccc2016/wp-content/uploads/2016/01/Experience-Driven-Design-of-Creative-Systems.pdf>.

Matthew John Yee-King. An Automated Music Improviser Using a Genetic Algorithm Driven Synthesis Engine. In Mario Giacobini, editor, *Applications of Evolutionary Computing*, number 4448 in Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2007. ISBN 978-3-540-71804-8 978-3-540-71805-5. URL http://link.springer.com.proxy.lib.sfu.ca/chapter/10.1007/978-3-540-71805-5_62.

Matthew John Yee-King. *Automatic sound synthesizer programming: techniques and applications*. PhD thesis, University of Sussex, 2011. URL <http://core.ac.uk/download/pdf/2710683.pdf>.

Michael Young. NN music: improvising with a ‘living’ computer. In *International Symposium on Computer Music Modeling and Retrieval*, pages 337–350. Springer, 2007. URL http://link.springer.com/chapter/10.1007/978-3-540-85035-9_23.

Marcel Zentner, Didier Grandjean, and Klaus R. Scherer. Emotions evoked by the sound of music: Characterization, classification, and measurement. *Emotion*, 8(4):494–521, August 2008. ISSN 1528-3542. doi: 10.1037/1528-3542.8.4.494. URL <http://proxy.lib.sfu.ca/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=pdh&AN=2008-09984-007&site=ehost-live>.

Appendix A

Revive: An Audio-Visual Performance with Musical and Visual Artificial Intelligence Agents

KIVANÇ TATAR
PHILIPPE PASQUIER
REMY SIU

AS PUBLISHED IN CHI EA '18 EXTENDED ABSTRACTS (ARTCHI) OF THE 2018 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (PP. 1-6). MONTREAL, CANADA: ACM PRESS. [HTTPS://DOI.ORG/10.1145/3170427.3177771](https://doi.org/10.1145/3170427.3177771)

REVIVE: An Audio-Visual Performance with Musical and Visual Artificial Intelligence Agents

Kivanç Tatar

Simon Fraser University
Vancouver, BC, Canada
ktatar@sfu.ca

Abstract

REVIVE explores the affordances of live interaction between the artificial musical agent MASOM, human electronic musicians, and visual generation agents. The Musical Agent based on Self-Organizing Maps (MASOM) has memorized sound objects and learned how to temporally structure them by listening to large corpora of human-made music. MASOM is then able to improvise live interacting with the other (human) performers by imitating the style of what it reminds it of. For each musician, a corresponding visual agent puts its sound and musical decision into images thus allowing the audience to see who does what. This reveals the musical gestures that are so often lost in electronic music performance. For CHI, MASOM plays with two live performers for a 20 minute audiovisual REVIVE experience.

Remy Siu

Gold Saucer Studio
Vancouver, BC, Canada
remysiu@gmail.com

Author Keywords

Artificial Intelligence; Live Performance; Audio-Visuals; Musical Agents; Multi-agent Systems; Generative Art; Computational Creativity; Musical Metacreation.

ACM Classification Keywords

J.5 [Computing Applications: Arts and Humanities]

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright held by the owner/author(s).
CHI'18 Extended Abstracts, April 21–26, 2018, Montréal, QC, Canada

ACM 978-1-4503-5621-3/18/04.
<http://dx.doi.org/10.1145/3170427.3177771>

Generative Art and Computational Creativity

Generative Art incorporates autonomous procedures in its making. The autonomous procedures used in Generative Art range from heuristics, that are often random or simple probabilistic procedures, to autonomous models that learn from a set of sample examples of outcomes (a corpus).

These autonomous procedures of Generative Art implement artistic creative tasks. However, not all creative tasks are artistic. In that sense, the scientific field of Computational Creativity (CC) researches all creative tasks including artistic tasks, such as producing music, and non-artistic tasks such as creating a new culinary recipe. Two applied sub-fields of CC are Metacreation and Musical Metacreation (MuMe) [4]. Metacreation is to endow machines with creative behavior whereas MuMe studies the partial or complete automation of musical creative tasks. REVIVE is an art project that integrates Metacreation and MuMe systems to live Audio-Visual (AV) performances.

REVIVE: Project Description

REVIVE is an experimental electronic music project featuring Kvanç Tatar, Philippe Pasquier, Remy Siu, and MASOM. MASOM is a musical agent, an artificial intelligence (AI) architecture for live performance [6]. A sub-field of MuMe, musical agents are artificial agents that automate musical creative tasks. Together, the three sonic performers and three visual agents produce a live performance of experimental electronic music, electroacoustic music, musique concrète, soundscape, through structured improvisation [7].

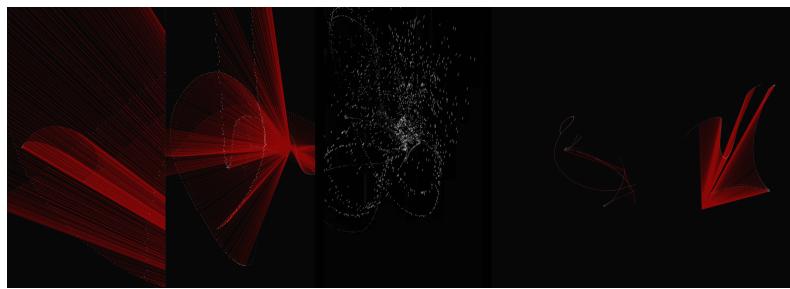


Figure 1: Five still images of visual agents generated by visual agents

on a large corpus of experimental music and electroacoustic music. MASOM extracts high-level features such as eventfulness, pleasantness, as well as timbral qualities to analyze and 'understand' the musical forms. Through its listening, the agent learns sound objects and how they are organized in human-made music.

The architecture of MASOM proposes an innovative perspective by combining a state of the art sound organization algorithm with pattern recognition algorithms (Figure 2). The musical agent creates a sound memory through automatic audio segmentation and thumbnailing, using audio features of timbre, loudness, fundamental frequency, duration, and music emotion features of eventfulness and pleasantness. The architecture applies Self-Organizing Maps [2, 3], a neural network machine learning algorithm. The agent organizes sounds on a two-dimensional map so that similar sound clusters locate closer to each other [1]. MASOM learns the temporality of musical form by applying pattern recognition on the organized sound memory. The agent assumes the musical form as temporal shifts on sound clusters that are organized in the feature space. During the live act, MASOM listens to the performance and locates its current state in the feature space. Using the previously learned temporal change patterns in the feature space, the agent generates sonic gestures to interact with other performers.

MASOM's architecture creates new artistic possibilities to be explored. Using MASOM, we could train a musical agent on the recordings of composers so that musicians could perform with the musical agent. Ideally, we could train a musical agent on any recording. In REVIVE, several MASOM agents are trained on various corpora of experimental electronic music including acousmatic music, glitch, intelligent dance music (IDM), and noise music. Acousmatic compositions use electronic means to create or process

MASOM stands for Musical Agent based on Self-Organized Maps. It is a machine improvisation software for live performance. The agent listens to itself and other musicians to decide in real time what to play next. MASOM is equipped with the latest algorithms in machine listening and is trained

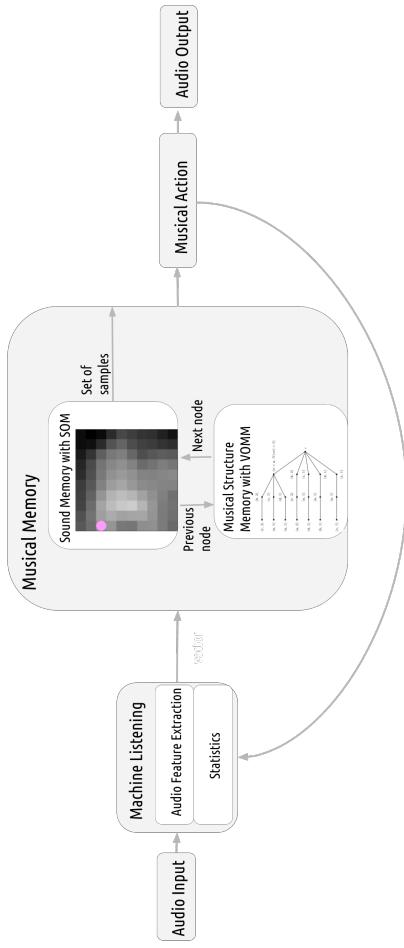


Figure 2: The system architecture of MASOM.

sounds to produce compositions. In acousmatic concerts, the audience listens to speakers in mostly dim-lit or darkened concert halls. Glitch music explores the idea of using sounds that are generated by the failure of any procedure. For example, using computers, glitch composers and performers overload the CPU to generate clicks and drops on the audio output. IDM composers use any sound object to produce dance music, extending their audio palette to unconventional sounds. Glitch sounds such as clicks, short impulsive noises frequently appear in IDM compositions. Noise music stands on the louder and aggressive end of the musical composition continuum. Noise music employs loud sounds to stimulate the body. The stimulations can be an ear pain caused by the loud sounds or pulsations generated by loud bass frequencies to vibrate the human body. MASOM learns from the recordings of fixed media pieces. The agent training process transforms these fixed record-

ings into interactive, performing agents. Charged with this knowledge and sonic memory, the musical agent then emulates the style of a composer. REVIVE celebrates music through iconic sonic textures.

Three musicians perform in this project: Kivanç Tatar, MASOM, and Philippe Pasquier (Figure 3). REVIVE improves the audience's perception of sonic gestures using visual cues. Three visual agents visualize the actions of audio agents (human performers and MASOM). The visual agents are generative and informed by the decisions made by sonic agents. Visual agents can perceive the action of sonic agents using a machine listening algorithm with high-level music features such as eventfulness and pleasantness as well as low level features such as spectral features, loudness, and pitch related features. The visual agents also perceive if an audio agent is active and when an audio agent initiates sound. Using perception abilities, the visual

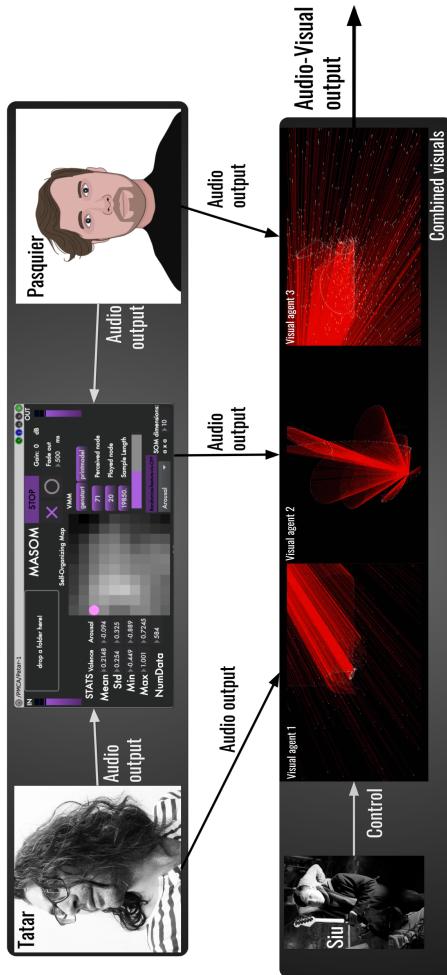


Figure 3: The performance setup of REVIVE, including MASOM and the three visual agents.

agents emphasize the actions of audio agents and make it easier for the audience to comprehend the connection between sonic gestures and the sonic agents' actions.

Previous Performances of MASOM

MASOM has performed in nine venues between October 2016 and January 2018. For the first performance of MASOM in Vancouver, BC, Canada, called *A Conversation with AI* [5], MASOM was trained on an [improvised noise music album](#) of Kivanç Tatar. During the freely improvised performance, MASOM acted as a clone of Tatar's improvisation style.

The second performance of MASOM was a collaboration of Metacreation Lab and the [New Orchestra Workshop \(NOW Society\)](#) in Vancouver, BC, Canada. The collective concert, called [madMethod](#), included acoustic instruments of piano,

drums, double bass, trumpet, saxophone, and electronics; as well as live visuals. MASOM was trained on the previous performances of NOW Society Ensemble for this performance.

The third performance of MASOM was a trio of Tatar and two MASOM agents. As in *A Conversation with AI*, both MASOM agents learned from the noise album of Tatar.

The performance was an act in the collective concert *Take the A/D Train* by İstanbul based noise collective [A.I.D.](#) The fourth performance included the same trio, and presented within the collective concert [RE/UN-SOLVED](#) in Vancouver, BC, Canada.

The fifth performance was the first performance of the project [PATAR](#), by Kivanç Tatar, MASOM, and Philippe Pasquier. For this project, the agent was trained on a corpus of elec-



Figure 4: A scene from the performance at the Deep Space 8k at the Ars Electronica Festival 2017.

troacoustic music. The act (in the collective concert *Barely Constrained* in Vancouver, BC, Canada) was an exploration of improvised acousmatic music. The sixth performance of MASOM (and the second performance of *PAT&R*) was a part of *Musical Metacreation Concert 2017* in Atlanta, Georgia, USA.

The seventh, eighth, and ninth performances of MASOM were collaborations of the Metacreation Lab and two Istanbul based media art companies: Ouchhh and *Audiofil*.

The first performance of this collaboration was performed three times at the *Ars Electronica 2017 Festival* in Linz, Austria (Figure 4); and MASOM was trained on the previous compositions of Mehmet Ünal from *Audiofil*. The second performance of this collaboration was presented as a part of *iMapp Bucharest 2017*. For this performance, MA-

SOM learned from the Audio-Visuals (AV) compositions of Ouchhh and *Audiofil*, and generated both audio and visuals. The generated AV was mapped on the façade of the Palace of Parliament in Bucharest, Romania. This collaboration ended with the performance at the *Circle of Light Festival* in Moscow, Russia. MASOM generated AV on the façade of the Bolshoi Theater.

The research and development of MASOM continue and the authors plan to present new versions and iterations of MASOM in future projects, collaborations, festivals, and exhibitions.

BIOs

Kivanc Tatar is a musician playing trumpet and electronics, a composer interested in experimental music, and a researcher studying artificial intelligence on the applications of music. As of 2017, his work has been exhibited in Germany, Italy, Romania, Austria, Brazil, Australia, USA (New York and Atlanta), Canada (Vancouver and Montreal), and Turkey; including the events Mutek_IMG, the cultural program at Rio Olympics 2016, and the Ars Electronica Festival 2017 (with the theme Artificial Intelligence). Currently, he is a Ph.D. candidate at the **School of Interactive Arts and Technology, Simon Fraser University**. In the **Metacreation Lab**, he is working on musical performance with artificial intelligence, **Musical Metacreation**, audio synthesis, audio programming, machine learning, generative art, and musical composition.

Philippe Pasquier works on creative processes and generative systems. He is both a scientist specialized in artificial intelligence, a multidisciplinary artist, an educator, and a community leader. His contributions range from theoretical research in multi-agent systems, computational creativity and machine learning to applied artistic research and

practice in digital art, computer music, and generative art. Philippe is an associate professor in the School for Interactive Arts + Technology at Simon Fraser University.

Remy Siu 薦逸南 is a composer and new media artist based in Vancouver, BC. Recently, his work has involved the construction of automated and variable performance apparatuses that employ light, sound, software, and the body. He is interested in creating friction and stakes between the performer, the interface, and the System through the use of game mechanics and failure. His output spans chamber music, dance, theatre, installations, and audio-visual work. He actively creates with **Hong Kong Exile** (interdisciplinary arts company) and **Manaila Patterson-O'Brien** (choreographer), and has worked with **Vicky Chow**, **Turning Point Ensemble**, **Quatuor Bozzini**, **Centre A Gallery**, **Pi Theatre**, **Theatre Replacement**, the **Western Front**, and others.

Acknowledgements

We would like to thank Laura Raveling for the proofreading. This work was funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Social Sciences and Humanities Research Council of Canada (SSHRC).

REFERENCES

1. Arne Eigenfeldt and Philippe Pasquier. 2010. Real-Time Timbral Organisation: Selecting samples based upon similarity. *Organised Sound* 15, 02 (Aug. 2010), 159–166. DOI: <http://dx.doi.org/10.1017/S1355771810000154>
2. Teuvo Kohonen. 1982. Self-organized formation of topologically correct feature maps. *Biological cybernetics* 43, 1 (1982), 59–69. DOI: <http://link.springer.com/article/10.1007/BF00337288>
3. Teuvo Kohonen. 1998. The self-organizing map. *Neurocomputing* 21, 1–3 (1998), 1–6. DOI: [http://dx.doi.org/10.1016/S0925-2312\(98\)00030-7](http://dx.doi.org/10.1016/S0925-2312(98)00030-7)
4. Philippe Pasquier, Arne Eigenfeldt, Oliver Brown, and Shlomo Dubnov. 2017. An Introduction to Musical Metacreation. *Computers in Entertainment* 14, 2 (Jan. 2017), 1–14. DOI: <http://dx.doi.org/10.1145/2930672>
5. Ash Tanasiychuk. 2016. A Conversation with Artificial Intelligence | VANDOCUMENT. (2016). <http://vandocument.com/2016/11/a-conversation-with-artificial-intelligence/>
6. Kivanç Tatar and Philippe Pasquier. 2017a. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*. Atlanta, Georgia, USA.
7. Kivanç Tatar and Philippe Pasquier. 2017b. MASOM: Musical Agent based on Self Organizing Maps. (2017). <http://metacreation.net/masom/>

Appendix B

Respire: A Breath Away from the Experience in Virtual Environment

MIRJANA PRPA
KIVANÇ TATAR
THECLA SCHIPHORST
PHILIPPE PASQUIER

AS PUBLISHED IN CHI EA '18 EXTENDED ABSTRACTS (ARTCHI) OF THE 2018 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (PP. 1-6). MONTREAL, CANADA: ACM PRESS. [HTTPS://DOI.ORG/10.1145/3170427.3177771](https://doi.org/10.1145/3170427.3177771)

Respire: a Breath Away from the Experience in Virtual Environment

Mirjana Prpa
 School of Interactive Arts+Technology
 Surrey, Canada
 mppa@sfu.ca

Kivanç Tatar
 School of Interactive Arts+Technology
 Surrey, Canada
 ktatar@sfu.ca

Thecla Schiphorst
 School of Interactive Arts+ Technology
 Surrey, Canada
 thecla@sfu.ca

Abstract
Respire is a virtual environment presented on a head-mounted display with generative sound built upon our previous work *Pulse Breath Water*. The system follows the changes in user's breathing patterns upon which it generates changes in the audio and virtual environment. The piece is built upon mindfulness-based design principles with a focus on breath as a primary object of the user's attention, and employs various approaches to augmenting breathing in the virtual environment.

Author Keywords
 Virtual Environment; Mindfulness; Interactive Art; Musical Agents

ACM Classification Keywords
 H.5.1. [Information Interfaces and Presentation (e.g. HCI)]:
 Multimedia Information Systems – Artificial, Augmented, and Virtual Realities

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

Copyright held by the owner/author(s).
CHI'18 Extended Abstracts, April 21–26, 2018, Montréal, QC, Canada
 ACM 978-1-4503-5621-3/18/04.
<https://doi.org/10.1145/3171042.3180282>

Introduction
Respire brings together three components: a virtual environment (via head-mounted display: Oculus Rift/HTC Vive), embodied interaction (via a respiration sensor), and an intelligent musical agent to listen to breathing patterns and generate the sound with affective properties (figure 2).

In *Respire*, a user journey starts from being immersed in a large body of water: an ocean. The ambiguous environment reacts to the user's breathing patterns and invites the user to create their experience by playful exploration of breathing patterns. Users, by becoming aware of their breath and their agency in the environment via breath, become co-creators of the experience. As time progresses and in respect to a variety of breathing patterns, the user is taken on the journey to different atmospheres and parts of the environment. The design principles we built *Respire* upon follow nuances of mindfulness-based designs, and ambiguity in design, and as such it encourages a user to re-connect with their breath and shift their attention inwards, towards bodily sensations of their breath while opening a space for reflection and exploration of self and the virtual environment.

The narrative of the experience of the virtual environment depends on the user's breathing and changes in their breathing patterns. The system captures the user's breathing frequencies via thoracic and abdominal breathing sensors, and does three things. First, the system determines the user's position in the environment: when the user breathes in, they rise in the VE, and then slowly sink (underwater) when breathing out, moving across the environment at a slow pace. Second, the system sends the breathing frequencies to the artificial musical agent that generates the audio in real-time by mapping the frequency of the user's breathing to the eventfulness of the audio. Affective computing in sound is implemented in the system design to allow the agent to estimate the affective qualities of generated sound and of the sounds in the memory of the agent. Affective audio accompanies the moody atmospheres of an impermanent and ever changing virtual environment that follows the journey of the user's breath. Finally, variety in breathing patterns trigger different elements in the VE, pop-

ulating the VE as the time progresses. For instance, fast thoracic breathing will trigger unpredictability of the environment and the visual elements displayed will convey sense of unrest and tensed atmosphere. In contrast, slow thoracic coupled with slow abdominal breathing will stabilize the environment in the state of serenity and peaceful atmosphere.

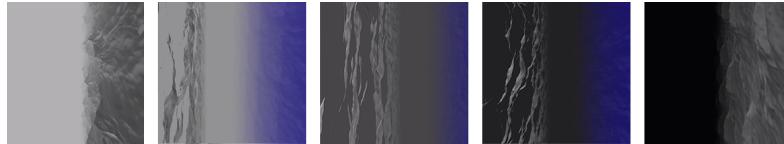


Figure 1: The changes in sky color over time: top image shows sky at minute 1, last image is at minute 6

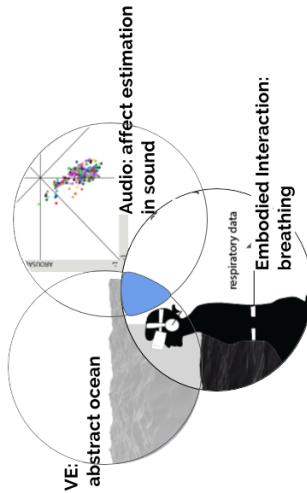


Figure 2: Respire's audio corpus in affect grid space.

Interaction Scenario
All the events in the environment are determined by the user's breathing patterns. Deep slow breathing triggers a particular part of the environment. Sustained breaths allow for staying in a specific place, while fast, strong breaths cause erratic movement. Mapping of a movement allows for interaction that is easy to understand: on the participant's inhale, the position of the participant rises in the environment, and on the exhale they sink, just like when submerged in water. As the time spent in the environment increases, the environment becomes more complex, with

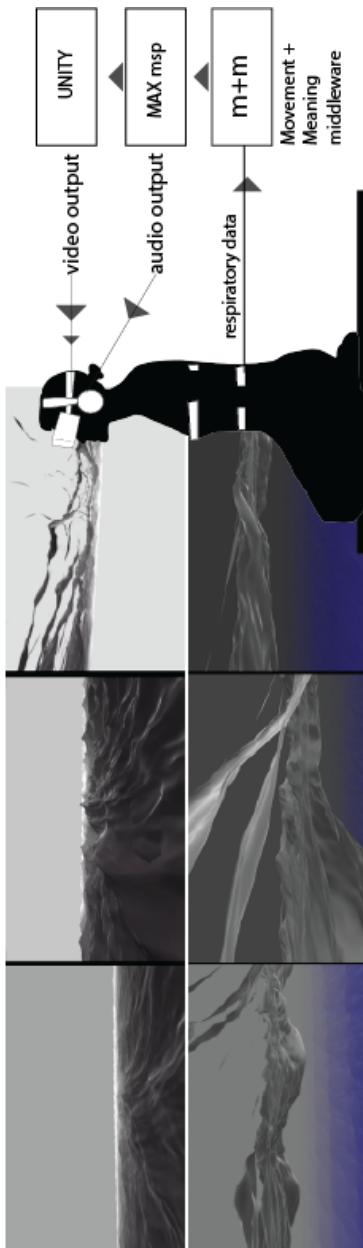


Figure 3: Respire: system design

visual components filling the space above and below the water. In our VE, the eventfulness of the audio is mapped to the appearance of the elements in the VE; for example, more eventful audio reflects in a more disturbed ocean surface and increased waves, or erratic movement of a flock of birds. The element that depicts the passage of time is the sky that changes the color from light gray to pitch black within a span of 6 minutes (see figure 1), however sustained slow breathing can slow this progression to up to 10 minutes.

System Description

The overall system outline is represented in Figure 3. One respiration sensor (Thought Technology) [2] attached to the user's abdominal area streams respiration data to M+M middleware [1] to a MAX patch¹.

Virtual environment

The virtual environment consists of a number of elements that are triggered by changes in breathing patterns. A user is immersed in an environment that is abstract but still perceptible as a representation of a mass of water – ocean. The scene is split into 3 sections: above the water (high arousal), in the waves (neutral arousal), and under the surface (low arousal). Each of these three sections represent different affective qualities. Rapid breathing patterns reflect highly aroused states, positioning the user above the water level. Uneven, fluctuating breathing positions the user in the middle of the waves. Finally, slow paced, deep breathing positions the user under the water surface level with the aim to elicit relaxed, calm, feelings in the user. These 3 sections convey different atmospheres in regard to audio/visual elements. From a water surface that is stormy and an atmosphere that is moody during rapid breathing, to a calm

underwater atmosphere that is surreal and ambient, breath

¹a visual programming framework for creative applications: <https://cycling74.com/products/max>

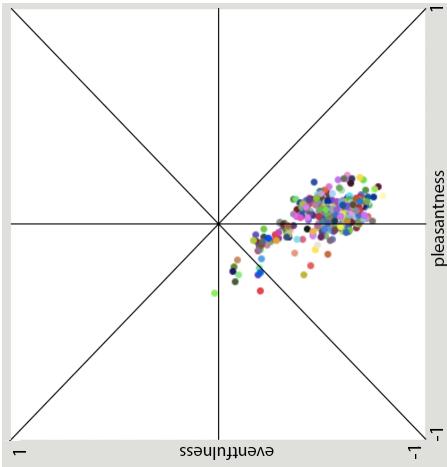


Figure 4: a segment of *Respire*'s audio corpus in affect grid space, centered around neutral pleasantness and low eventfulness.

is the key interaction that happens between a user and the environment. By changing the breathing frequencies and switching between chest and thoracic breathing, the user changes the scene, and then consequently, the scene might influence the user. This influence is anticipated by allowing the user to make a connection between their breathing and the elements they see in the environment (e.g., sustained, slow paced breathing “makes” the environment look very different).

Audio environment

The audio is generated by an autonomous agent that selects samples from the audio corpus according to the frequency of the user's breathing. All audio samples are tagged with different eventfulness and pleasantness properties us-

ing a state of the art music emotion recognition algorithm [4, 3]. The average affect values of each audio sample in *Respire*'s audio corpus are presented in the figure 4. We created the audio corpus by recording two, three, four, and five voice chords with quartile harmony on the piano. On that corpus we applied pitch shift and time stretch to produce more sounds around neutral valence and neutral to low arousal in the affective space. The user's abdominal breathing patterns are extracted using the wavelet transform of the breathing data and mapped to the highest and the lowest arousal (eventfulness) values in our audio corpus. Hence, we map the frequency of the user's breathing to the eventfulness of the audio. The overall affect of the audio is centered around neutral arousal with the aim of creating a particular audio aesthetic. Our goal was to lead our users towards relaxing states, by introducing audio low in arousal (in audio vocabulary of affect: eventfulness), and staying in neutral to the positive end of valence axis (pleasantness).

The overall eventfulness and pleasantness values of the audio environment are sent to the 3D game engine Unity 3D² along with respiration data via OSC messages. This data generates visual changes in the VE presented to the user via HMD. The user listens to the audio environment with circumaural noise-cancelling headphones.

Creating the experience from a breath

The elements in the virtual environment are triggered by a combination of two values: 1. breathing frequencies from thoracic and abdominal sensors, and 2. eventfulness and pleasantness values from the audio environment. The combinations of 1. and 2. are depicted in figure 5. The figure shows the affect grid (similar to the grid in figure 4) and the visual environment is determined by the combination of the

²a game engine: <https://unity3d.com/>

aforementioned values. For instance, fast thoracic breathing causes the environment to switch to a moody, dark atmosphere, positioning the user in the upper left quadrant of the affect grid (high eventfulness, low pleasantness). By engaging in fast abdominal breathing, additional elements are added to the scene, such as a disturbed, wavy surface of the ocean. In contrast, slow thoracic and abdominal breathing will position a user on the surface of the ocean, letting them float gently on it, while the virtual environment slowly gets brighter and vivid in color. The color of the entire environment is determined by the audio values: pleasantness determines the color of the sky (low pleasantness = dark sky), whereas eventfulness determines how vivid the colors are (the more eventful the audio = brighter colors of the virtual environment).

Design Principles

Respire is built with the intention to help users reconnect with an embodied experience, often lost in our interaction with new and emerging technologies. To guide the user's attention inwards to the self, we focused on mindfulness-based design and breathing as an object of the user's attention. The changes in *Respire* come directly from changes in breathing patterns, allowing the user to become aware of their breath and the agency they have in the environment. Designing to support breath awareness was our primary focus in interaction design. To support further reflection and self-exploration of breath, we present a user with an ambiguous environment. In our design, breathing is augmented through various audio and visual cues, out of which a narrative emerges through embodied interaction.

User Experience

In this piece, the musical agent listens to the user's breathing and takes the user on a journey through abstracted worlds with different affective qualities creating a user-

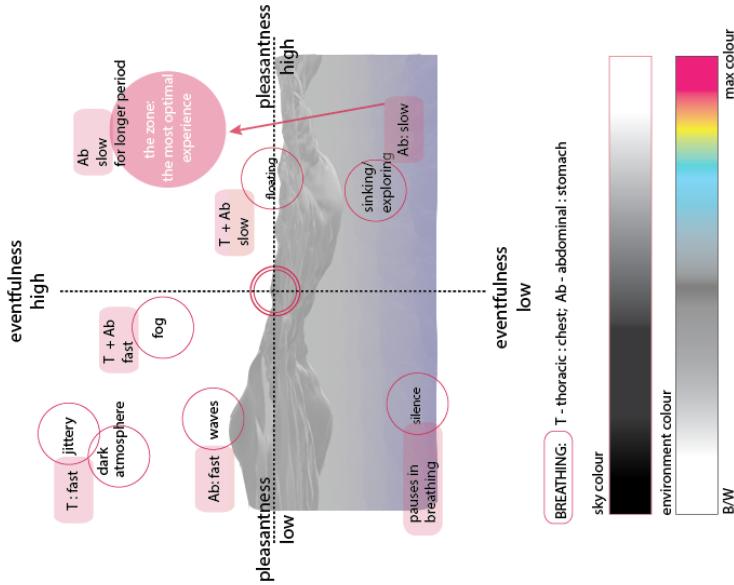


Figure 5: *Respire's* audio corpus in affect grid space.

dependent narrative. From the calm to stormy oceans and ambiguous architectures that one is immersed in and that elicit curiosity; this is a journey within one's own breathing. The environment allows each audience member to create their alternate realities based on the interaction between two dynamic systems: the user and the system. This makes each journey unique, personal, and unrepeatable.

Acknowledgements

We thank all the colleagues and friends who provided unlimited testing and advices. We would like to acknowledge Jianyu Fan for his help on the audio affect estimation model.

We thank Ash Tanasiychuk for tireless proofreading. The authors would like to acknowledge that this work would not be otherwise possible without grant support from the Social Sciences and Humanities Research Council of Canada (SSHRC), and equipment support from Natural Sciences and Engineering Research Council of Canada (NSERC).

REFERENCES

1. 2015. Movement and Meaning | Canarie Middleware. (March 2015). <http://www.mplusm.ca/>
2. 2015. Thought Technology Ltd. ProComp2 - 2 Channel Biofeedback & Neurofeedback System w/ BioGraph Infiniti Software Thought Technology Ltd. (April 2015). <http://thoughttechnology.com>
3. Jianyu Fan, Miles Thorogood, and Philippe Pasquier. 2016. Automatic Soundscape Affect Recognition Using A Dimensional Approach. *Journal of the Audio Engineering Society* 64, 9 (2016), 646–653.
4. Kivanc Tatar and Philippe Pasquier. 2017. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*. Atlanta, Georgia, USA.

Appendix C

Automatic Synthesizer Preset Generation with PresetGen

KIVANÇ TATAR
MATTHIEU MACRET
PHILIPPE PASQUIER

AS PUBLISHED IN JOURNAL OF NEW MUSIC RESEARCH, 45(2), PP 124–144, 2016,
DOI.ORG/10.1080/09298215.2016.1175481

Abstract

PresetGen addresses the target preset generation problem - given a sound and a synthesizer, finding a preset that best approximate the target sound - in the case of real-world synthesizer OP-1. The OP-1 consists of several synthesis blocks, and it is not fully deterministic. We propose and evaluate a solution to preset generation using a multi-objective Non-dominated-Sorting-Genetic-Algorithm-II. Our approach makes it possible to handle the problem complexity and returns a small set of presets that best approximate the target sound by covering the Pareto front of this multi-objective optimization problem. Moreover, we present an empirical evaluation experiment that compares performance of three human sound designers to that of *PresetGen*, and shows that *PresetGen* is human competitive.

Keywords: Artificial Intelligence; Genetic Algorithms; Sound Synthesis; Multi-objective Optimization; Evolutionary Computation

1 Introduction

Modern musicians, composers and sound designers have been using audio synthesizers to imitate the sound of acoustic instruments electronically or to explore new sounds. Synthesis techniques have been developed to craft musical sounds and sound effects.

Efficient control and exploration of a synthesizer's sound space typically requires expert knowledge of the related synthesis technique, which mixes theoretical and empirical knowledge. It is common that composers have to abandon their task of making music to concentrate on the task of programming a synthesizer, i.e. tuning parameters to create the desired sound. Depending upon the synthesis technique used, a synthesizer user needs to tune a variable number of parameters. Therefore, for a complex synthesizer, the size of the parameter search space can quickly become large and challenging to handle manually by the user. Exploring the parameter search space to

tune a synthesizer, even in a principled fashion, can become a time-consuming activity.

The synthesizer interface can present an obstacle between musical ideas and their expression to musicians who do not have this technical knowledge. The synthesizer's parameters used to craft the sound are specific to the particular synthesis technique being used, and rarely reflect the human understanding or perception of sound. This makes the synthesizer interface unintuitive and challenging to handle for someone interested in musical creativity rather than programming a synthesizer.

Synthesizer manufacturers often provide the user with a large number of parameters settings, also called presets. A synthesizer user can use presets to produce recognizable or interesting sounds. Presets are also starting points for users to explore the synthesis sound space. However, even if these presets are meant to convey the variety of sounds that the engine is capable of generating, they fail to cover the entire synthesis sound space. In this work, we focus on the problem of synthesizer preset generation to match a target sound. The ultimate goal is to provide this automatic system, *PresetGen*, to users so that they can generate their own presets for their target sounds without having to deal with obscure parameter settings.

A first step is the development of a process that can efficiently search the synthesizer parameter space to identify presets which approximate given target sounds. This work examines the use of evolutionary computation for preset generation problem of a modern commercial synthesizer developed by Teenage Engineering¹ the OP-1. The OP-1 is an all-in-one portable synthesizer, sampler, and controller. The OP-1 introduces additional challenges comparing to previously studied synthesizers (Horner, A. and Beauchamp, J., 1996; Schatter, G. and Züger, E. and Nitschke, C., 2005; Riionheimo, J. and Välimäki, V., 2003; Vuori, J. and Välimäki, V., 1993; Horner A. and Beauchamp J. and Haken L., 1993; Lai, Y. and Jeng, S.K. and Liu, D.T. and Liu, Y.C., 2006; Mitchell, T., 2012; Bozkurt, B. and Yüksel, K., 2011). OP-1 contains several synthesis engines, effects (FX)

¹<https://www.teenageengineering.com/products/op-1>

and Low-Frequency-Oscillators (LFOs), which make the parameter search space larger and more complex. Furthermore, the OP-1 is not fully deterministic such that the generated sound is slightly different for a given OP-1 preset.

In this study, we based our evaluation design on Johnson's recommendation about the experimental analysis of algorithms (Johnson, D., 2002). We especially focused on ensuring reproducibility and comparability.

Section 2 provides a literature review of previous works using evolutionary computation to search automatically the parameters of synthesizers embedding various synthesis techniques.

Section 3 presents general background about evolutionary computation. We describe the canonical genetic algorithm (GA) and then discuss the notion of the fitness landscape and its relationship to problem difficulty. We point out the limitations and challenges that can arise when using a GA to solve our synthesis preset generation problem. We present the extensions and algorithmic alternatives commonly used to deal with these challenges and limitations. Finally, we conclude on a variation of the canonical GA, which is developed to optimize multiple objective functions simultaneously. We select this variation - the Non-dominated Sorting Genetic Algorithm- II (NSGA-II) - in the design of our final system.

Section 4 describes the OP-1 synthesizer and analyzes the problem of generating synthesizer presets that match a given target sound. We also point out arising difficulties particular to the OP-1 synthesizer. In Section 5, we present the iterative methodology that we adopted to solve the problem.

Section 6, presents *PresetGen*'s system design. We consider several sub-problems of increasing complexity when adding more and more parameters to the search. We show how we modify a standard GA step by step to solve these sub-problems. Furthermore, we present how it leads us to switch to using a Non-dominated Sorting Genetic Algorithm-II that incorporates a three objective fitness function and a Gray code encoding to handle the problem in its full complexity.

Section 7 describes the sound collection that we used to study the performance statistics of *Preset-*

Gen. We explain why we have chosen a particular twenty four sounds in two categories, non-contrived and contrived sounds. Twelve of these sounds were non-contrived sounds whereas the other twelve were contrived sounds. Moreover, we explain why we decided to study both non-contrived and contrived sounds.

Section 8.1 shows the correlation between our fitness function and the distance between target preset and the matching preset for a contrived sound. Hence, we present that our fitness function is able to lead our system to the region in the parameter space where the target is located.

Section 8.2 presents *PresetGen*'s performance statistics by running *PresetGen* ten times for each target sound. First, we use contrived target sounds to assess the ability of *PresetGen* to retrieve the target synthesizer's parameters. The contrived sound experiments' results show that *PresetGen* successfully approximates to the target presets by obtaining low errors between target sounds and matching sounds for the FFT and STFT. *PresetGen* also performs very well to approximate the amplitude envelopes as shown by low errors in the envelope distance. Results also show discrepancies in performance depending upon the nature of the target sound, some inducing a harder optimization problem than others. The non-determinist nature of the OP-1 synthesizer made impossible the convergence to the exact target parameter values. The second part of Section 8.2 presents the performance statistics of *PresetGen* with non-contrived sounds. Our results show that the resulting Pareto fronts sounds perceptually similar to the target sounds. Our trials show that the optimization problems induced by non-contrived sounds are more difficult than the one induced by contrived sounds with higher distance values for the three objectives (Envelope distance, FFT distance, and STFT distance).

Section 9 presents the empirical evaluation of OP-1 preset generation for non-contrived sounds. This experiment shows that *PresetGen* can improve a preset generation task of non-contrived sounds in terms of quality. The experiment shows that *PresetGen* is

human-competitive in quality while it is not human-competitive in time complexity. However, we expect this problem to diminish with the developments in the microprocessor technologies and increasing computational processing power.

2 Evolutionary computation for sound synthesis

We start our literature review with particular optimization problems of relatively low complexity to more general problems of higher complexity, involving modern synthesizers that embed multiple complex synthesis engines.

The complexity of the preset generation problem can vary tremendously according to the number and the nature of the synthesis parameters to search. First efforts focused on optimizing a limited number of synthesis parameters. Horner, A. and Beauchamp, J. (1996) used additive synthesis to replicate instrument sounds. They used GA to find how many breakpoints they need to match a target sound on their piecewise-linear approximation of additive synthesis amplitude and frequency envelopes. The number of oscillators to use and their frequencies were not determined by the GA but through spectral analysis.

Chan, Yuen, and Horner (1996) implemented GA with discrete summation and hybrid sampling wavetable model synthesis methods to match a musical instrument tone. Hybrid sampling wavetable method uses sampling for attack portion of a sound, and wavetable synthesis for gradually changing sustain and release. This method can synthesize sounds with a dynamic spectrum using multiple wavetables. They used this synthesis method as spectral interpolation approach on the problem of matching a target sound. In this approach, target sound's different parts' spectrum crossfaded with each other to match the target sound. They used GA to determine the basis spectra and the best amplitude envelope for this spectrum.

Wakefield and Mrozek (1996) used subtractive synthesis to create artificial reverberation. A GA was used to search for low-order filter parameters so that

the generated impulse response best matched that of a target room transfer function.

Horner A. and Beauchamp J. and Haken L. (1993) also used GA to optimize several frequency modulation (FM) synthesis parameters: the modulation indices, carrier and modulator frequencies. The spectral error between the original and target spectra served as a fitness function in guiding the GA's search for the best FM parameters to mimic instrumental sounds. In our previous work, we used a similar technique to optimize modulation indices, carrier and modulator frequencies for Modified FM (ModFM) synthesis (Macret, M. and Pasquier, P. and Smyth, T., 2012).

Vuori, J. and Välimäki, V. (1993) applied GA to estimate each parameter of a non-linear physical flute model. Each chromosome of this system represented eight different parameters of the physical model. The spectral error between the original and matched spectra served as the fitness function. The algorithm converged smoothly and effectively towards the target sound.

Bozkurt, B. and Yüksel, K. (2011) conducted synthesizer preset generation experiments with genetic algorithms in application to multiple-modulator FM synthesis. Contrary to the FM synthesis systems previously presented (Macret, M. and Pasquier, P. and Smyth, T., 2012; Horner A. and Beauchamp J. and Haken L., 1993), their GA implementation included all parameters that control their FM synthesis method.

Mitchell, T. (2012) compared three algorithms, Multi-member Evolution Strategy (such as 1+4), Multi-start (1+1) Evolution Strategy and Clustering Evolutionary Strategy(CES), to generate presets for FM synthesizers. CES clusters the population at the beginning of each iteration using k-means clustering. This parent population re-clustering ensures that convergences on the same niche merge to form a single cluster. The author stated that this was beneficial in multi-modal search spaces. The author's implementation's synthesis architecture includes parallel FM synthesis blocks. The number of these blocks was also another parameter to be optimized in the author's system. This application used Short-Time Fourier Transform (STFT) in the fitness measure.

The author divided the evaluation of these algorithms to two, non-changing static tones, and time-varying dynamic sounds. Considering these three algorithms, the author concluded that CES performed the best in his experiments.

Yee-King, M. and Roth, M. (2008) used a GA to generate presets of Virtual Studio Technology instruments (VSTi) synthesizers to match a given target sound. Their implementation did not include VST FXs. Their system is called *SynthBot*. Although VSTi synthesizers includes more complex synthesis architectures than former studies, the OP-1's particular synthesis architecture introduces more complexity to the preset generation problem. We explain these new challenges in Section 4. *SynthBot* uses single objective of the sum squared error between the target sound's and the matching sound's Mel Frequency Cepstrum Coefficients (MFCCs) in its fitness function. Our implementation shows that the single objective approach gives unsatisfactory results in the case of OP-1. We present results of single objective fitness functions and our final multi-objective design in Section 6.

3 Background on evolutionary computation

Contrary to the canonical GA where a single near optimal solution is considered, multi-objective GAs usually return a set of solutions, known as *Pareto-optimal* solutions. A solution is said *Pareto-optimal* if none of its objective functions can be improved in value without impairment in some of the other objective values. A solution is *dominated* if some other solution is better for one or more objectives without being worse for the remaining objectives. The *Pareto front* is the set of solutions that are *Pareto-optimal* at a given moment. Non-dominated Sorting Genetic Algorithm-II (NSGA-II) (Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T., 2002) has become the standard approach solving multi-objective problems.

In the next subsections, we will define the notion of non-dominated sorting and crowding distance and show how they are used in the main loop of the

NSGA-II to converge toward an optimized Pareto front.

3.1 A non-dominated sorting approach

The goal is to sort the population in non-dominated fronts. We consider a population of size N and M objectives. Comparing each solution with every other solution in the population to find if it is dominated, we identify solutions of the first non-dominated front in the population. At this stage, all individuals in the first non-dominated front are found. We temporarily discount the solutions of the first front and repeat the above procedure. Thereby, we find the individuals in the next non-dominated front. We use the same procedure for finding third and higher levels of non-domination. Thus, the worst case is when there are N fronts, and there exists only one solution in each front. Hence, this algorithm requires an overall $\mathcal{O}(MN^3)$ computations. The non-domination level for a given individual is the rank of the front it belongs. NSGA-II uses a fast non-dominated sorting approach that requires only $\mathcal{O}(MN^2)$ comparisons.

3.2 Diversity preservation

Along with convergence to the *Pareto-optimal* set, we expect implemented GA to maintain a high spread of diversity in the solution set. NSGA-II assigns to every individual a *crowding distance* (Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T., 2002). Crowding distance calculation is as follows;

1. Sort the population along each fitness function objective. (For each fitness function objective, we end up with different rankings of the same population.)
2. For each fitness function objective, assign infinite distance value to the boundary solutions - the individuals with the smallest and the largest objective value along that objective.
3. Calculate the remaining individual's each fitness function objective's crowding distance. For each

fitness objective, x_i represents the corresponding objective value of the individual with index i in the sorted population. Calculate the individual's absolute normalized Euclidian distance of j th objective with the formula,

$$\frac{|d_j(x_{i-1}, x_i) - d_j(x_i, x_{i+1})|}{f_{j,max} - f_{j,min}} \quad (1)$$

where $f_{j,max}$ and $f_{j,min}$ refers to corresponding objective's maximum and minimum value in the population.

4. Assign a crowding distance to every individual by summing that individual's all fitness function objectives' crowding distances.

Equation 2 summarizes the crowding distance calculus for an individual in the population.

$$d(x_i) = \sum_{j=1}^{N_{obj}} \frac{|d_j(x_{i-1}, x_i) - d_j(x_i, x_{i+1})|}{f_{j,max} - f_{j,min}} \quad (2)$$

where N_{obj} is the number of objectives, $d_j(x_{i-1}, x_i)$ (resp. $d_j(x_i, x_{i+1})$) gives the normalized Euclidian distance value of the first (resp. second) adjacent solution when sorted in ascending order of magnitude for the objective i .

The introduction of crowding distance provides that the boundary solutions are more likely to be kept in the next generation (see following section for more details).

3.3 Main loop

We present the NSGA-II procedure in Figure 1. First, a population P_0 of size n_{pop} is randomly initialized. The fitness of every individual is then evaluated using the fitness function. The population is sorted based on non-domination using *non-domination level*, is assigned to every individual. Using the algorithm described in Section 3.2, a *crowding distance* is also calculated for every individual.

Contrary to the canonical GA, the selection operator used in NSGA-II is a tournament selection operator based on dominance between two individuals. If

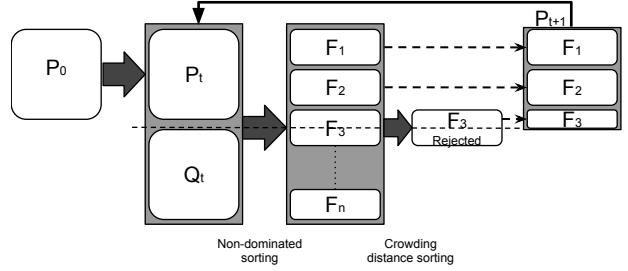


Figure 1: NSGA-II procedure (Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T., 2002)

the two individuals do not inter-dominate, the selection is made based on crowding distance. The recombination and mutation operators are used to create an offspring population Q_0 of size n_{pop} . A combined population $P_0 \cup Q_0$ is considered. This population is sorted according to non-domination. Since all previous and current population individuals are included in $P_t \cup Q_t$, *elitism* is ensured. Solutions belonging to the best non-dominated set F_1 are from the combined population and must be emphasized more than any other solution in the combined population. If the size of F_1 is smaller than n_{pop} , we definitely choose all members of the set for the new population P_{t+1} . The remaining members of the population P_{t+1} are chosen from subsequent non-dominated fronts in the order of their ranking. This procedure is continued until no more sets can be accommodated. Say that the set F_n is the last non-dominated set beyond which no other set can be accommodated. In general, the count of solutions in all sets from F_1 to F_n is larger than the population size n_{pop} . To choose exactly n_{pop} population members, we sort the solutions of the last front by *crowding distance* in increasing order and choose the best solutions needed to fill all population slots.

4 The OP-1 Synthesizer

The OP-1 is an all-in-one portable synthesizer, sampler and controller developed by Teenage Engineering (TE)² and illustrated in Figure 2. Figure 3 shows an

²<https://www.teenageengineering.com/products/op-1>

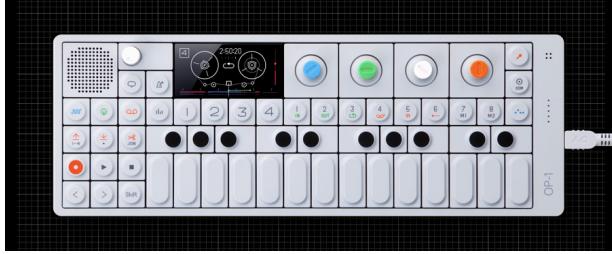


Figure 2: OP-1 picture

overview of the OP-1’s synthesis architecture. TE provided us with a C++ library that embeds most of the functionalities of the OP-1. We had access to seven different synthesizer engines (FM, Digital, DrWave, String, Cluster, Pulse and Phase), four different FX (Delay, Grid, Punch, Spring) and three different LFOs (Tremolo, Value, Element). In the following sections, the parameters selecting the engine, FX and LFO will be referred to as *type parameters*. Only one engine, one effect and one LFO can be used at a given time to produce a sound. An ADSR envelope is also always applied to the sound. Once chosen, the synthesizer engine, FX, LFO and ADSR can be each controlled individually by the 4 knobs. In the following sections, the parameters controlling the knobs will be referred to as *knob parameters*. The knob parameters are mapped to integers ranging from a minimum of 0 to a maximum of 32767, corresponding to the fine-tuning mode of the OP-1. The OP-1 has twenty-four physical keys and it is possible to change the octave from -4 to 4. Therefore, a hundred and twenty different keys ($8 \times 12 + 24$) are available when using the OP-1. We refer to a set of OP-1 *knob parameters* and OP-1 *type parameters* as an OP-1 preset. More details about the OP-1 can be found on the Teenage Engineering website³.

Equation 3 gives the number of different possible combinations.

$$N_{eng} \times N_{LFO} \times N_{FX} \times N_k^{N_{knobs} \times N_t} \times N_{keys} \quad (3)$$

where N_{eng} is the number of engines type, N_{LFO} the number of LFO types, N_{FX} the number of FX type,

³<https://www.teenageengineering.com/products/op-1>

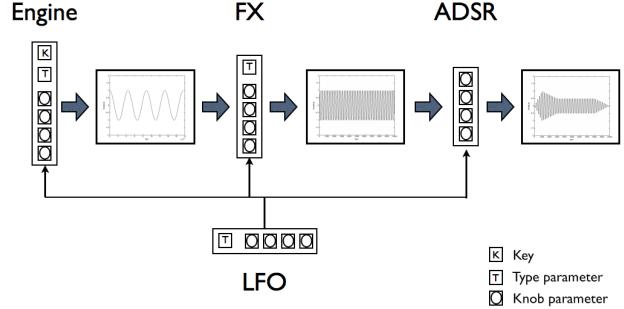


Figure 3: The OP-1’s modular synthesis architecture

N_t the number of modules that can be controlled by knobs (engine, LFO, FX and ADSR), N_k the number of possible integer values for each knob and N_{keys} the number of keys. Their numerical values are given in Table 1. An estimate of the total number of possible combinations for the OP-1 synthesizer is then 10^{76} .

Searching the synthesizer parameters space to generate a preset that can match a given target sound has all the characteristics of a real-world problem. First, the search is very large (10^{76} possible different combinations). By comparison, the number of atoms in the observable universe is estimated at 10^{80} . Second, the synthesizer, OP-1, is not fully deterministic. The output sound is slightly different with the same set of input parameters, which induces noise in the evaluation and can then slow down or even mislead the search. We conducted two experiments to prove and measure the non-determinism of the OP-1 and results are available online (Tatar, K. and Macret, M. and Pasquier, P., 2015). Third, the search space is multimodal. For example, for a given individual, switching from an engine to another completely changes the nature of the output sound. As a result, its fitness objectives values also substantially change causing a discontinuity in the fitness landscape. It also completely modifies the mapping of the *knob parameters*. For example, the *knob parameters* for an FM engine

N_{eng}	N_{LFO}	N_{FX}	N_k	N_{knobs}	N_t	N_{keys}
7	3	4	32767	4	4	120

Table 1: Synthesizer parameters complexity

do not map to the same synthesis parameters than the *knobs parameters* for a Digital engine. Finally, our trials showed that there are a large number of local minima (see Section 8.1.1). For instance, it is often possible to get a similar level of sound approximation using two different engines. Given these problem characteristics, it is not conceivable to use a random search or a simple optimization technique such as hill climbing or greedy algorithms to find a good set of parameters to match a given target sound. These techniques are highly dependent on the initial conditions and do not scale well to large and difficult search spaces (Roth, M. and Yee-King, M., 2011).

5 Methodology

As described in Section 3, GAs are search algorithms that mimic the process of natural evolution. GAs are especially well adapted to the characteristics of our problem. First, GAs scale very well to the large and complex search space induced by the OP-1. Contrary to gradient search methods, they are less susceptible to converge prematurely to a local optimum (Rocha, M. and Neves, J., 1999).

Second, GAs also perform well in search spaces where the evaluation is approximative or noisy (Jin, Y. and Branke, J., 2005), as is the case with the OP-1 and its non-fully deterministic output. Adjustable selection pressure makes it possible to keep diversity in the population. A large number of individuals are evaluated for each generation. Because mutation and crossover are stochastic operators, it is common for an individual to be rediscovered several times during the evolution. The fact that *PresetGen* re-evaluates a re-discovered individual each time, reduces the effect of the noise in the evaluation due to the non-determinism of OP-1.

GAs are complex algorithms with a large set of parameters to tune (population size, stopping criteria, choice of the genetic operators). We explored several options to find the best configuration for the GA. In the following sections, we refer to the target sounds generated using the OP-1 as *Contrived sounds* (Mitchell, T. and Creasey, D., 2007). Using *Contrived sounds* as target sounds has two advan-

tages. First, it ensures that a solution exists. Second, we can measure the performance of the algorithm by calculating its distance to the optimal solution.

We adopted an iterative design process and considered problems of increasing complexity. Table 2 describes these problems ordered by increasing complexity. From problem 1 to 4, we progressively added the knob parameters. In this first set of problems, we fixed the type parameters to limit the search space discontinuities. Finally, in problem 5, all type and knob parameters were searched. At first, we limited the search to the four knobs controlling the engine parameters (Problem 1). We experimented with different GA configurations until we found one configuration able to either, in the best-case scenario, reverse engineer the target set of OP-1 parameters or, in the worst-case scenario, gave a perceptually satisfying approximation of the target sound. Once a satisfying GA configuration was found, the four knobs controlling the ADSR were added (Problem 2). The previously satisfying GA configuration was tested on the new problem. If this configuration was not satisfying anymore, we adjusted the GA parameters again until a satisfying one was found. This process was reiterated with the other problems until we obtained good performance when searching every parameter (Problem 5).

In the following subsections, we describe the final system implementation of *PresetGen* for searching all the parameters. We explain our design choices given the observations gathered during the different steps of our iterative design process.

6 System design

6.1 Representation

At first, we decided to encode these parameters using a binary representation. Using a mixed-integer or real-value representation did not make sense in our case because both type and knob parameters in the OP-1 are integers and not real values. Moreover, when using a mixed-integer or real-value representation, the crossover operator loses its ability to explore the genotype space as it is just exchanging integer

Pb. Id	Engine		FX		Key Octave	LFO		ADSR	N_{bits}
	Type	Knobs	Type	Knobs		Type	Knobs		
1		✓							60
2		✓						✓	120
3		✓		✓				✓	180
4		✓		✓			✓	✓	240
5	✓	✓	✓	✓	✓	✓	✓	✓	257

Table 2: Problem description

or real parameters between chromosomes. Instead, with a binary representation, the crossover operator exchanges bits, which can lead to some changes in the related OP-1 parameters values.

Our trials using the binary representation on Problem 1 (see Table 2) seemed to indicate that the GA was always converging to the same local minima. Investigating further, we realized that 11 bits would have to be changed to go from 4095 (011111111111) to 4096 (100000000000) to improve the best individual fitness and the objective fitness values, which is very unlikely to happen. Then, we switched from a binary encoding to a Gray code encoding for both *type parameters* and *knob parameters*. The Gray code is based on the idea that two successive values differ by only one bit. Our trials with this new encoding showed that the GA now converged toward the target set of parameters. Thus, we chose to keep this representation for our system. Our chromosome is made up of blocks representing the type and knobs parameters. The two first lines of Table 2 in bold letters show the final chromosome design.

For the full problem complexity (Problem 5), each OP-1 preset is represented by a string of 257 bits. The number of distinctly possible bit strings is then $2^{257} = 10^{257 \log_{10}(2)} \approx 10^{77}$. In Section 4, we calculated that the number of distinct possible OP-1 presets has an order of magnitude of 10^{76} . The difference of a factor 10 between these two orders of magnitude can be explained by the fact that, when the number of values to encode is not a power of 2, the binary encoding encodes for more values than necessary. For example, we have to encode 120 different keys in our chromosome (see Table 1). With 6 bits, it is possi-

ble to encode $2^6 = 64$ different keys and with 7 bits, $2^7 = 128$ different keys. We, then, chose to use 7 bits and applied a scaling function to keep the decoded integers between 0 and 119. However, this difference of 9 between the number of keys to encode and the number of possible distinct bit strings when using 7 bits does explain the difference we observed in the orders of magnitude.

6.2 Genetic operators

Losing diversity during the evolution is a normal phenomenon given that we apply a selection pressure on the population. However, a lack of diversity can lead to premature convergence because there is not enough genetic material to explore the fitness landscape. One reason for the loss of diversity is the recombination of identical chromosomes. Indeed, when two strictly identical chromosomes are selected for cross-over, two offsprings identical to their parents are produced. This phenomenon causes the diversity to go down. To avoid this situation, we apply a crossover operator that tests the parent chromosomes before recombining them. If they are identical, the first offspring will be a copy of the parents and the second offspring will be a new randomly generated chromosome. This simple technique is shown to be efficient in slowing down the diversity loss and prevent premature convergence (Rocha, M. and Neves, J., 1999). *PresetGen* uses a two-point crossover and a crossover rate of 60 %.

The mutation operator participates in both exploration and exploitation (local search). Flipping one bit in a Gray code can either lead to a small change

Table 3: Mutation examples

Integer	Binary	Gray
4095	011111111111	01000000000000
4096	10000000000000	11000000000000
2048	01000000000000	01100000000000

in the coded parameter (local search) or a relatively large change in the coded parameter (exploration: by jumping to another area of the fitness landscape). Table 3 shows two examples of mutation using the same bit strings than in Section 6.1. Flipping one bit in the Gray code can either lead to a single increment in the integer value (4095 to 4096) or lead to a large change in the integer value (4095 to 2048). This flip-bit mutation operator is applied to every individual in the population whether crossover is applied or not. In our system, the probability of flipping k bits in a N_{bits} long chromosome follows a binomial law with $p = \frac{1}{N_{\text{bits}}}$ and $n = N_{\text{bits}}$.

The population’s diversity is critical for the success of the GA. We use following approaches to measure the diversity:

- The proportion of different individuals for each generation,
- The numbers of each module types for each generation,
- The knob parameters standard deviation for each generation,
- The distance standard deviation for each generation.

Our experiments showed that our approach maintains a high level of diversity. Detailed information about these approaches as well as diversity statistics of our experiments can be found online (Tatar, K. and Macret, M. and Pasquier, P., 2015).

6.3 Fitness function

In our attempt to solve Problem 1 in Table 2, we used the Euclidian distance between the Short-Time

Fourier Transform (STFT): d_{STFT} . Equation 4 shows the Euclidian distance between the STFT for the individual t and c .

$$d_{\text{STFT}}(t, c) = \sqrt{\sum_{i=1}^{N_w} \sum_{j=1}^{N_s} (t_{i,j} - c_{i,j})^2} \quad (4)$$

The sampling rate was 44100 Hz and we set a window size N_s of 1024 samples (23 ms) and an overlapping of 512 samples (11.5 ms). N_w is the total number of windows. Our trials showed that this fitness function worked well when we restricted the optimization to include only the four knobs which controlled the engine parameters (Problem 1).

However, when we added the four knobs controlling the ADSR parameters (Problem 2), the GA appeared to converge prematurely. A further investigation of this phenomenon showed that the weight of the amplitude envelope in the Euclidian distance between the STFTs is significant. For example, consider a target sound T and two candidate sounds A and B (see Figure 4). A has a similar spectrum to the spectrum of T for the first short-time windows but not for the last ones because the amplitude envelope for A has a shorter release time than the one for T . Globally, the spectrum for B is not as similar to the spectrum for T than A but their amplitude envelope is the same. The weight of these last short-time windows (shown by the rectangle in Figure 4) can make B appear closer to T than A according to the Euclidian distance between the STFT described in Equation 4.

In this context, a correct set of engine knobs parameters (A) can be discarded because the associated ADSR knobs parameters are not right. It slows down the evolution because some good genetic material is lost. It can even lead to premature convergence if this set of engine knobs parameters is never recovered again later in the evolution.

It is not surprising that, in previous work (Macret, M. and Pasquier, P. and Smyth, T., 2012; Horner A. and Beauchamp J. and Haken L., 1993), the envelope was determined analytically for every individual in the population. However, it is not possible to do this in our case. Indeed, we know that, for the ADSR,

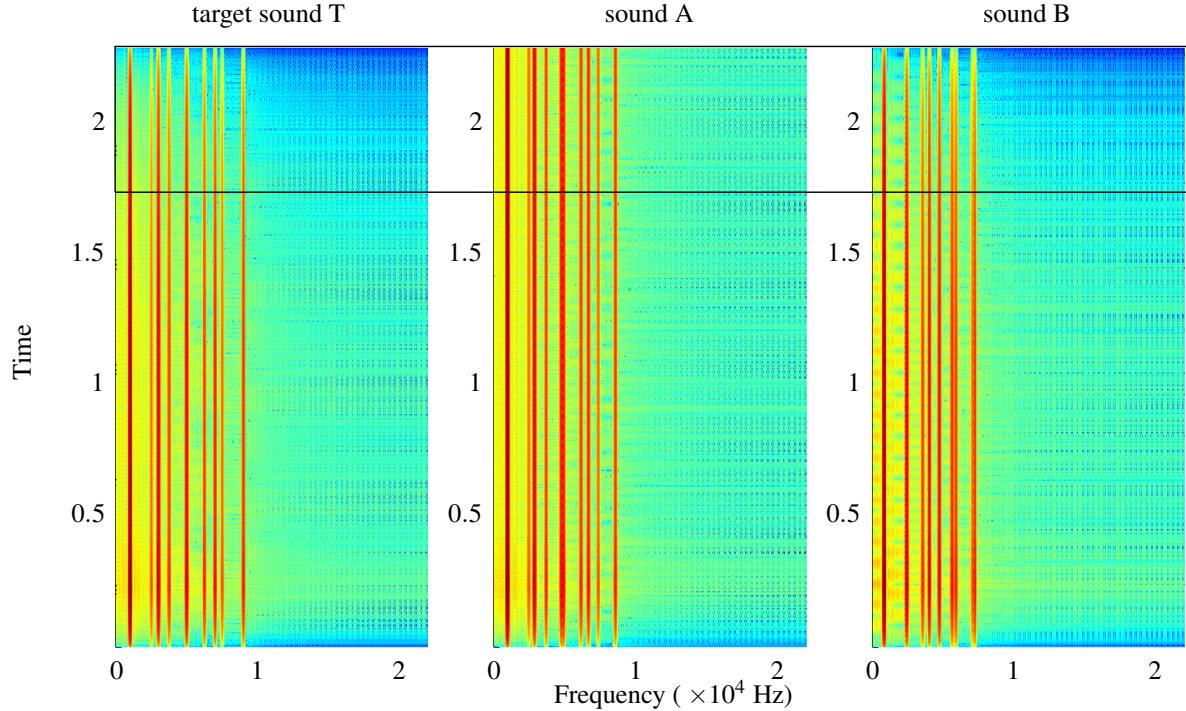


Figure 4: Weight of the envelope in the Euclidian distance for the STFT

the first knob is mapped to the Attack, the second one to the Decay, the third one to the Sustain and the last one to the Release but we do not know the precise value mapping between these knob parameters and the embedded ADSR values. Moreover, an LFO can be used to modulate one or several of these ADSR knobs. Another idea would be to perform a local search to set the ADSR knobs parameters for each individual in the population. However, it would be computationally expensive. Furthermore, given the non deterministic nature of our synthesizer, any classic local search algorithm such as Greedy algorithm or Hill climbing would likely fail.

We decided to uncouple the amplitude envelope from the spectral components as much as possible to avoid the premature convergence observed with Problem 2. Thus, we chose to extract two separate sound features: 1) the FFT computed on the entire sound;

and 2) the amplitude envelope. Computing the FFT on the entire sound mitigates, to some extent, the effect of the amplitude envelope on the spectrum. We extracted the amplitude envelope using the Hilbert transform followed by a low-pass filter.

Our first idea was to put these two sound features in an aggregate fitness function (see Eq. 5). However, it is challenging to choose the appropriate weights for the amplitude envelope a_{env} and the FFT a_{FFT} to make the system converge.

$$f = \frac{a_{\text{env}} f_{\text{env}} + a_{\text{FFT}} f_{\text{FFT}}}{a_{\text{env}} + a_{\text{FFT}}} \quad (5)$$

Therefore, we chose to consider two objectives: FFT and amplitude envelope instead of only one: the STFT. We implemented this new two-objectives fitness function in a multi-objective framework, the Non-dominated-Sorting-Genetic-Algorithm-II. Trials

showed that this new system converged to the target set of parameters for Problem 2.

However, when we added the four knobs controlling the LFO or FX (Problem 3-4), our system was converging prematurely again. The explanation was that the addition of an LFO or FX made the spectrum of the target sound non-stationary. The FFT on the entire length of the sound was not able to capture the variation of the spectrum over time. Then, we added back the STFT as a third objective to deal with this limitation. Contrary to simple GA, the NSGA-II uses a selection operator based on non-domination sorting. Therefore, contrary to the simple GA using STFT as fitness function, an individual with a good set of engine knobs parameters would be more likely kept in the population even if it has wrong ADSR knobs parameters. Indeed, this individual would have a high fitness value for the FFT and a low fitness value for the envelope. It would be then kept in the population because it is dominating the population according to the FFT objective. In the canonical GA, this individual would likely be discarded because its fitness value would be affected by a wrong amplitude envelope.

6.4 Selection

PresetGen is based on a Non-dominated Sorting Genetic Algorithm II (NSGA-II). Deb's work (Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T., 2002) and Section 3 present details about this algorithm.

PresetGen uses a population of five hundred individuals for each generation. This number of individuals was empirically determined as a good trade-off between performance and computational cost.

6.5 Stopping criteria

The optimization process terminates if the weighted change in the three objective fitness, given by Eq. 6, is less than 10^{-10} over 200 generations. δ_n is the weighted change at generation n , f_k is the best objective fitness score at generation k , $N = 200$ if $n \geq 200$ otherwise $N = n$. If this condition is never verified, the optimization process stops after 3000th generations.

$$\delta_n = \sum_{i=1}^N \left(\frac{1}{2}\right)^{N-i} (f_{n+1-i} - f_{n-i}), \quad (6)$$

6.6 Pareto front

The Pareto front is the set of non dominated individuals for the three objectives.

In our first trials, the Pareto front was very large at the end of the evolution (more than two thousand individuals). Upon closer examination, we realized that strictly identical individuals were present in the Pareto front. Crowding selection is only made on objective fitnesses because the synthesizer is not fully deterministic and the non-domination. Then, we added a similarity rule that tests whether the chromosome of an individual is already present before adding it to the Pareto front. This simple rule made it possible to cut the size of the final Pareto front by a factor of more than two.

However, the size was still too large (around a thousand individuals) to be easily analyzed by a user. Further investigating the Pareto front, we realized that a large number of individuals sounded perceptually identical even if they were produced using different sets of parameters. We refined the similarity rule to limit this kind of duplicate individuals in the Pareto front. We stated that two individuals are considered identical if they have the same engine/LFO/FX types, identical key/octave and the Euclidian distance between the knob parameters is less than a thousand (3 % of the knob parameter range). This value was determined by making tests on a large numbers of Pareto fronts. Depending on the target sound, this last change made it possible to reduce the size of the Pareto front to between ten and a hundred and fifty individuals while conserving its quality and diversity. A hundred and fifty individuals in the Pareto front is still very large to be handled by a user. We applied a technique developed by Chaudhari et al. (Chaudhari, M. and VDharaskar, R. and Thakare, V., 2010) to select the most significant individuals in the Pareto front when its size is superior to ten individuals. This approach consists of the following steps:

1. Apply a k -means clustering algorithm to cluster the solutions enclosed in the Pareto front. We implement clustering on the OP-1 presets to help the user to identify the OP-1 presets with different sound engines in the Pareto front.
2. Determine the optimal number of clusters, k . The silhouette (Rousseeuw, P., 1987) of an individual is a measure of how closely it is matched to other individuals within its cluster and how loosely it is matched to individuals of the neighbouring cluster. A silhouette $s(i)$ close to 1 implies that the individual i is in an appropriate cluster, while $s(i)$ close to -1 implies that i is in the wrong cluster. Thus the average $s(i)$ of the entire Pareto Front is a measure of how appropriately the Pareto Front has been clustered. A value of the average silhouette is obtained for several values of k with $k < 10$. The k that gives the highest average silhouette width is selected. Figure 5 shows an example of a silhouette plot. Each bar represents silhouette value ($s(i)$) of an individual in the Pareto front. Y axis represents the number of clusters given to the k -means algorithm.
3. For each cluster, select a representative solution. For each cluster, the individual, within the cluster, that encodes the OP-1 presets that is the closest to the cluster centroid presets is selected as the representative solution.
4. Analyze the results. At this point, the user can analyze the k representative solutions of the clusters and then explore the individuals of the cluster that seems the most promising.

6.7 Full problem complexity

In Problem 5, we add the type parameters. These extra parameters to search induces multimodality to our problem (see Section 4). However, our trials show that adding the type parameters to the search (Problem 5) do not diminish the final solution quality when we compare them to final solutions found for Problem 4. The *right* type parameters are determined

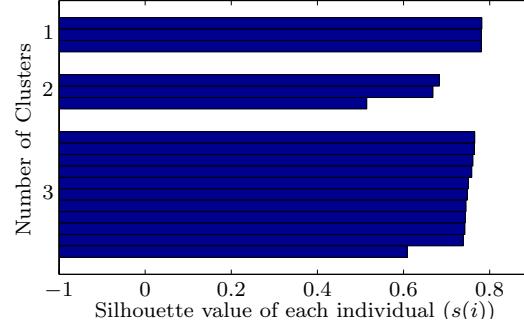


Figure 5: Example of a silhouette plot

in early generations and become prominent in later generations. Then, the evolution continues as if it would be Problem 4 being solved. This phenomenon is induced by the selection pressure and mimics well the behaviour of a human asked to perform the same task. One would broadly explore the possibilities of the synthesizer and quickly select an engine and key, after which one would fine tune the knob parameters (see Section 4).

6.8 Implementation

In the implementation of *PresetGen*, we used the OP-1's C++ library provided by *Teenage Engineering*. We utilized DEAP (Distributed Evolutionary Algorithms in Python) Python framework for the GA implementation (Fortin, F. and De Rainville, F. and Gardner, M. and Parizeau, M. and Gagné, C., 2012). We conducted audio feature extraction using Python bindings of YAAFE (Yet Another Audio Feature Extractor) library (Mathieu, B. and Essid, S. and Fillon, T. and Prado, J. and Richard, G., 2010). We ran our Python code on Bugaboo computing cluster that is part of Westgrid Compute Canada (*Westgrid - Compute Canada*, n.d.). We used Matlab for post-processing with k -means clustering (MATLAB, 2011).

7 Sound collection

Using *contrived sounds* - i.e. sounds actually generated by the OP-1 - as target sounds allows us to validate *PresetGen*'s system design, making it possible to show that *PresetGen* can reverse engineer the parameter setting (or presets) of a given target sound generated by the OP-1. Given the complexity of the algorithm and its running time, we chose to limit our evaluation to twelve contrived sounds. We selected these sounds to have a sample of spectrums that was diverse and representative of the OP-1 possible outputs. We especially focused on having diversity in spectral variation, noisiness, and spectral spread. The spectral variation $S_v(t)$ (or spectral flux) represents the amount of variation of the spectrum along time. It is computed from the normalized cross-correlation between two successive amplitude spectrum $a(t-1)$ and $a(t)$.

$$\begin{aligned} S_v(t) &= \frac{\sum_k (a_k(t) - a_k(t-1))^2}{\sqrt{\sum_k a_k(t-1)^2} \sqrt{\sum_k a_k(t)^2}} \\ S_v &= \sum_t S_v(t) \end{aligned} \quad (7)$$

$a_k(t)$ is the k^{th} bin of the amplitude spectrum at time t . Our set of non-contrived sounds consists of both the sounds with a stationary spectrum and the sounds with a dynamic spectrum, as measured by their respective spectral variation.

We made sure that we used each engine, LFO and FX at least once to generate this set of sounds. These contrived sounds are available to listen online (Tatar, K. and Macret, M. and Pasquier, P., 2015). Table 4 described these contrived sounds. For each of these sound, we also give its overall spectral variation S_v (Peeters, G., 2004).

We distinguish two groups: the sounds with a stationary spectrum ($S_v < 10^{-5}$) and the sounds with a dynamic spectrum ($S_v \geq 10^{-5}$).

A second evaluation was performed on twelve non-contrived sounds including synthetic sounds, instrument sounds, a male voice sound and a cat sound. These sounds were carefully chosen to have a good diversity in spectral variation, noisiness, and spectral spread.

We used recorded target sounds as non-contrived sounds. As with the contrived sound, we distinguish two groups: the sounds with a stationary spectrum ($S_v < 10^{-5}$) and the sounds with a dynamic spectrum ($S_v \geq 10^{-5}$).

8 Results

8.1 The Detailed Statistics of two sounds

In this section, we present the detailed statistics of two runs with two target sounds, *conf4* and *cat*. *conf4* exemplifies a contrived sound whereas *cat* exemplifies a non-contrived sound. In the following section 8.2, we present the global statistics of multiple runs with all target sounds. The statistics and results for all runs are available online (Tatar, K. and Macret, M. and Pasquier, P., 2015).

8.1.1 Objective fitness

The goal of the GA is to minimize the three objectives fitnesses (Envelope Euclidian distance, FFT Euclidian distance and STFT Euclidian distance). Figure 6 shows the reduction of the FFT distance over the generations. On both graphs, a viewer can observe plateaus, also called *punctuated equilibria*, that can be interpreted as periodic improvements in fitness. One can also distinguish the *exploration* phase and *exploitation* phase on these graphs. The *exploration* phase happens at the early stage of the evolution with a quick and significant improvement in the fitness. The *exploitation* phase follows with episodic and small improvement in the fitness. Exploration also continues during exploitation. The length of these phases can vary depending of the nature of the target sound. For example, a viewer can see that the *exploration* phase is around 50 generations for the contrived sound and around 350 generations for the non-contrived sound. Another interesting point to note is the difference in the range of the final objective fitnesses for the contrived sound trials and for the non-contrived sound trials. For example, the final FFT fitnesses vary between 0 and 500 for the con-

Conf. Id	Engine	FX	LFO	key	octave	S_v	Stationary
0	FM	Grid	No	17	4	1.252×10^{-4}	N
1	Digital	Delay	No	1	-1	7.641×10^{-6}	Y
2	Cluster	Delay	Value	16	3	6.625×10^{-4}	N
3	Digital	Punch	Tremolo	16	3	1.172×10^{-4}	N
4	Digital	Delay	No	9	2	9.301×10^{-6}	Y
5	FM	No	No	0	2	3.055×10^{-4}	N
6	FM	No	No	11	1	7.128×10^{-5}	Y
7	String	No	No	20	-3	3.055×10^{-4}	N
8	Cluster	No	No	12	0	6.740×10^{-6}	Y
9	Pulse	No	No	9	-1	2.279×10^{-4}	Y
10	Phase	No	No	9	-4	1.154×10^{-4}	N
11	Phase	Punch	Element	16	1	9.4187×10^{-6}	Y

Table 4: Contrived sounds

trived sound trials and between 2000 and 3000 for the non-contrived sounds trials. This difference illustrates the fact that, as expected, it is more difficult to approximate non-contrived sounds (large final objective fitnesses) than contrived sounds (smaller final objective fitnesses).

8.1.2 Distance/fitness correlation

Another way of tracking the improvements of the system for the non-contrived sounds is to look at the parameter distance to the target parameters over the generations. We considered two distances: the Euclidian distance and the Hamming distance.

We define the Euclidian distance between two individuals in Equation 8.

$$d_e(r, s) = \sqrt{\sum_{i=1}^N (r(i) - s(i))^2} \quad (8)$$

where r and s are two individuals set of parameters. N is the total number of parameters. $r(i)$ (esp. $s(i)$) are the i^{th} parameter of individual r (resp. $s(i)$).

The Hamming distance between the two bitstrings of chromosomes is defined in Equation 9.

$$d_h(u, v) = \frac{c_{01} + c_{10}}{n} \quad (9)$$

where c_{ij} is the number of occurrences of $u[k] = i$ and $v[k] = j$ for $k < n$, n being the number of bits. We

study the hamming distance as well because *PresetGen* uses bitwise representations of OP-1 parameters to find a contrived sound target's preset.

Figure 7 shows the minimum of these distances to the target individual/chromosome over the generations for the *conf4* sound. Contrary to the objective fitnesses (Figure 6), the curves are not monotonic decreasing. The Euclidian distance even appears to increase over the generations. Possible explanations are that there are several local minima or even several possible solutions. Then, it is possible to achieve a good approximation of the target sound using completely different parameters than the ones used initially to produce it. For this reason, tracking these distances is of limited relevance to evaluate the performance of *PresetGen*. However, they make it possible to estimate the difficulty of our problem observing the correlation between fitness and distances.

We study the correlation of the distance - between the target preset genotype and an individual's genotype - and the fitness value to evaluate the difficulty of the problem being solved.

To further study the difficulty of our problem, we plot in Figure 8 the average FFT distance to the target sound per generations against the average Euclidian and Hamming distances of phenotypes (OP-1 presets) to the target preset. One can see that, in average, the Euclidian and Hamming distances are well correlated to the fitness with a correlation coefficient equals to 0.67 for the Euclidian distance, and 0.70

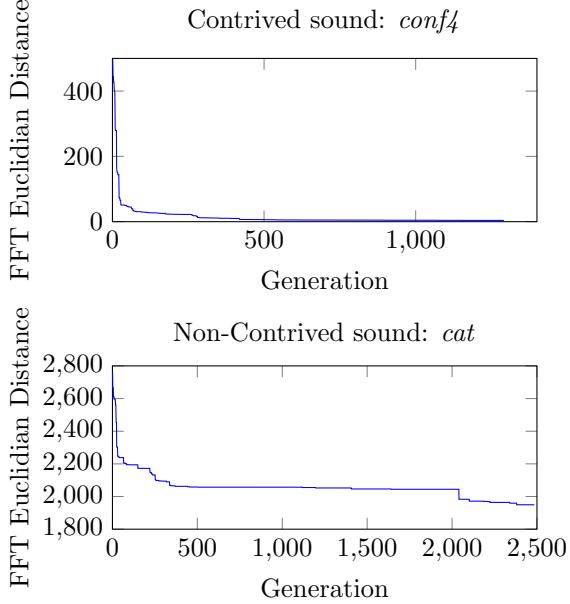


Figure 6: Objective fitness: minimum FFT Euclidian Distance to the target sound in the population

for the Hamming distance. It shows that the fitness lead, in average, the GA to the region in the parameters space where the target is located. We can also observe, with the cluster of points around the coordinates $[3 \times 10^4, 0]$, that a single large optimal plateau exists.

We also plot in Figure 9 the fitness (FFT distance to the target sound) of each individual against their phenotypes' (OP-1 presets') Euclidian and Hamming distances to the target preset for the last generation of the GA. One can see that as the distance to the target sound increases, the fitness decreases. This inverse correlation between distance and fitness makes the convergence toward the exact local or global minima very challenging and also explain why we never get the exact target presets at the end of the optimization.

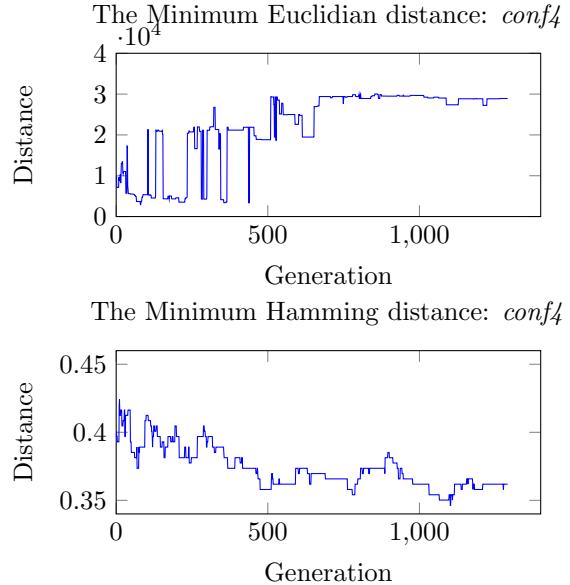


Figure 7: The minimum distance (in the population) between the target preset's genotype and the individual's genotype per generation

8.1.3 Pareto front Analysis and Clustering

Our trials show that the Pareto front is most of the time too big to be easily handled by the final user. The idea is to give a sample of individuals (less than ten) that is a good representation of diversity in the Pareto front. As described in Section 6.6, we apply a k-means clustering algorithm to cluster on the solutions enclosed in the Pareto Front. As the clustering is done on the OP-1 knob parameters, the user can easily retrieve all the individuals enclosed in the given cluster starting with the centroid individual. Indeed, we assume that going from the centroid individual to any individuals in the cluster, the user just has to slightly modify the centroid individual's parameters. Moreover, we also assume that individuals in the same cluster sound similar as their OP-1 configuration are similar.

The k-means clustering algorithm requires that the number of clusters k to be chosen before running the algorithm. The average silhouette of the data is an useful criterion for assessing the natural number of

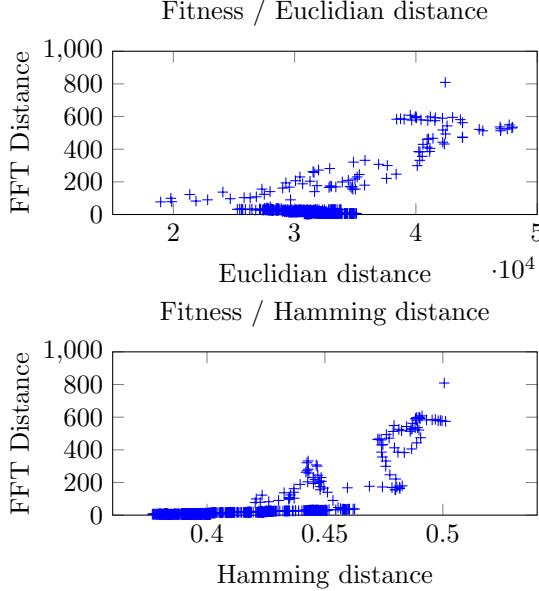


Figure 8: Fitness-Distance (global)

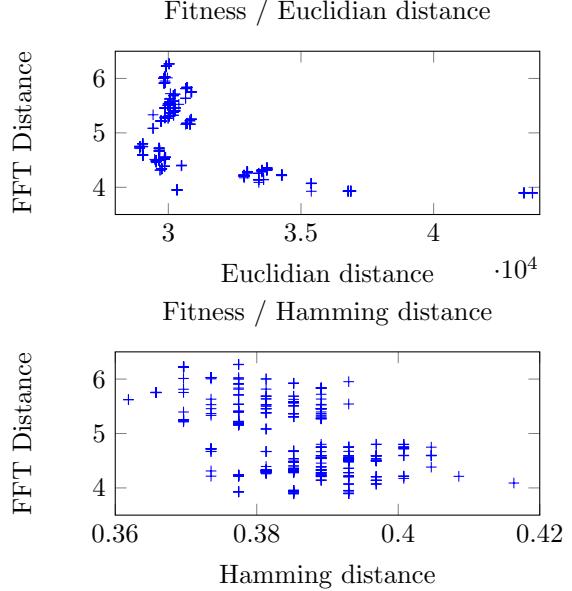


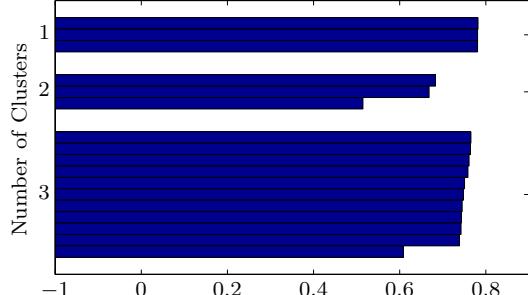
Figure 9: Fitness-Distance (last generation)

clusters. The silhouette of a datum is a measure of how closely it is matched to data within its cluster and how loosely it is matched to data of the neighbouring cluster, i.e. the cluster whose average distance from the datum is lowest. A silhouette close to 1 implies the datum is in the appropriate cluster, while a silhouette close to -1 implies the datum is in the wrong cluster. We run the k-means clustering algorithm several times with increasing values of k starting from 1 to a maximum of 10. We select the value of k that gives the best average silhouette. Figure 10(a) and Figure 10(b) respectively show the silhouettes for the *conf4* target and for the *cat* target sound that give the best average silhouette. Each bar represents the silhouette of an individual. One can observe these bars more easily on Figure 10(a) as the Pareto front is small for the *conf4* target sound. As one can see on both Figure 10(a) and 10(b), the average silhouette is close to 1 and there is no individual with a small or negative silhouette.

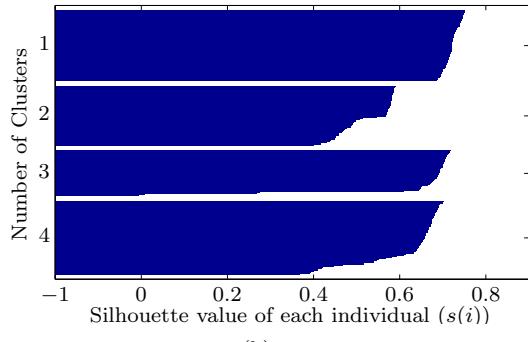
Figure 11(a) and Figure 11(b) give a 3D representation of the Pareto front over the three objectives (FFT, STFT and envelope). As described in

the previous section, the clustering is done on the OP-1 parameters and not on the objective fitness. However, a viewer can observe that the Pareto front is also well clustered over the three objectives. It makes sense because neighbours in the OP-1 parameters space should also be neighbours in the objectives space. However, we can observe some outsiders (for example, one individual belonging to the cluster 3 on Figure 11(a)). It can be because the clustering is not perfect, but it can also illustrate the inverse correlation between parameter distance and objective fitnesses we found in Section 8.1.2.

The individual in each cluster that is the closest to the centroid is chosen to represent its clusters. Figure 13(a) (respectively Figure 12(a)) shows a comparison between the centroid individual spectrogram and the related target sound spectrogram for the *conf4* sound (respectively *cat* sound). For the contrived sound *conf4*, the two spectrograms look perfectly similar. However, the objective values for the FFT and STFT on Figure 11(a) are small but not equals to zero. Hence, the match is not perfect but it is very close as it is not possible to make the difference per-



(a) *conf4*



(b) *cat*

Figure 10: Silhouettes

ceptually when we listen to the two sounds. For the non-contrived *cat*, the two spectrograms do not look as similar as in the case of the contrived sound. With a non-contrived sounds, it is not possible to know in advance if we can generate exact match of a non-contrived target sounds with the OP-1 synthesizer. However, we see strong similarities in the frequency range in both spectrograms and also in the spectral envelope. When listening to the sounds, you can also find obvious perceptual similarities. Figure 13(b) (respectively Figure 12(b)) shows a comparison between the centroid individual waveform and the related *conf4* (respectively *cat*) target sound waveform. For both contrived and non-contrived sounds, we can see that the amplitude envelopes are either close (*cat*) or look identical (*conf4*).

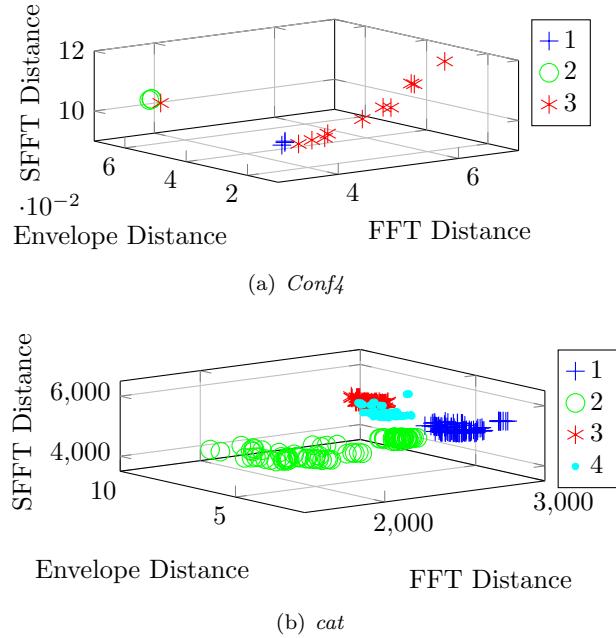


Figure 11: Pareto front

8.2 Statistics

8.2.1 Bootstrapping

We ran the algorithm at least ten times for each target sound. We used Bootstrapping to obtain estimates of summary statistics (Johnson, D., 2002). This method involves taking an original dataset, and sampling from it with replacement to form a new sample, called a bootstrap sample. A bootstrap sample has the same amount of data with the original data set. However, it is not identical to the original data set. We repeat and exclude data from original data set to create a bootstrap sample. We repeat the generation of a bootstrap sample for a large number of times (a thousand in our case). For each of these bootstrap samples, we compute the desired statistic that provides an estimate of the distribution of the descriptive statistics. This technique makes possible the extraction of more useful information when the sampling size is small, as is the case in our trials due to the time complexity of the problem.

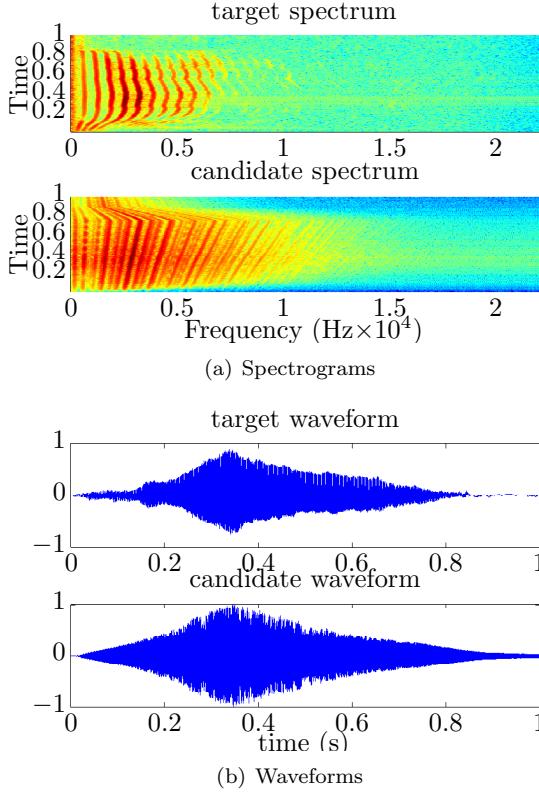


Figure 12: Non-contrived sound (*cat*): Centroid individual for the cluster 1

For each of the measures described above, we used Bootstrapping to get an estimate of its minimum, maximum, mean and standard deviation. We also used the bootstrap shift method test (Johnson, D., 2002) to assess the significance of every comparison we performed. This test has the advantage of being distribution-free and of scaling well with small sample size.

8.2.2 Measurement

In our experiments' descriptive statistics, we evaluated the solution quality and the running time.

The solution quality was measured differently for the contrived sounds and the non-contrived sounds. For the contrived sounds, we already know what are

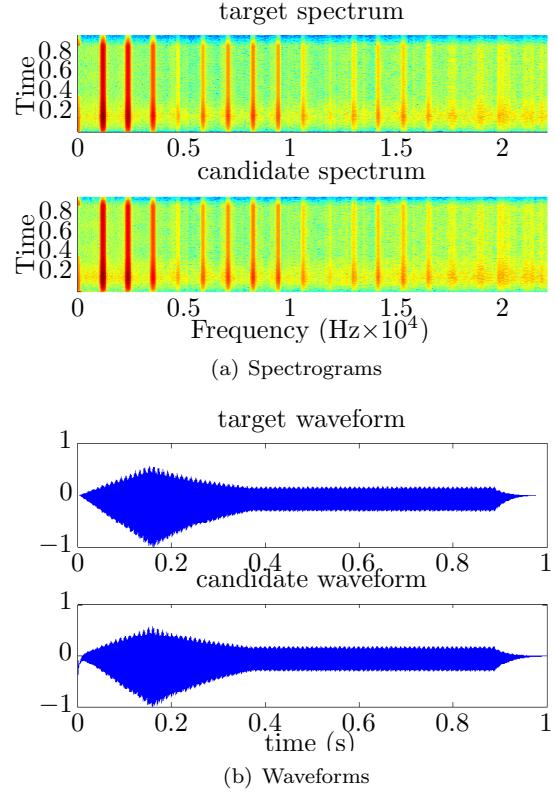


Figure 13: Contrived sound (*conf4*): Centroid individual for the cluster 1

the target OP-1 presets. However, OP-1 involves indeterminism in its synthesis architecture. An OP-1 preset does not generate the exact same sound each time. Therefore, we generate ten sounds using a target preset in addition to the target sound, and compare them to the target sound. With a determinist synthesizer, their objective fitness values (FFT, envelope and STFT) would be equals to zero, but it is not the case with the OP-1. For each objective, we define the best possible objectives value F_b as the minima of the objective fitness value over these ten sounds. For each objective, we calculate the error Δ_r for each run subtracting this best possible objective values F_b to the best objective fitness value obtained in the particular run f_r (see Equation 10).

$$\Delta_r = f_r - F_b \quad (10)$$

For the non-contrived sounds, we are only able to measure the final fitness values for the three objectives at the end of the evolution. In both cases, the running times are measured by the number of generations before the GA reaches the stopping criteria ($nbGen$).

8.2.3 Contrived sounds

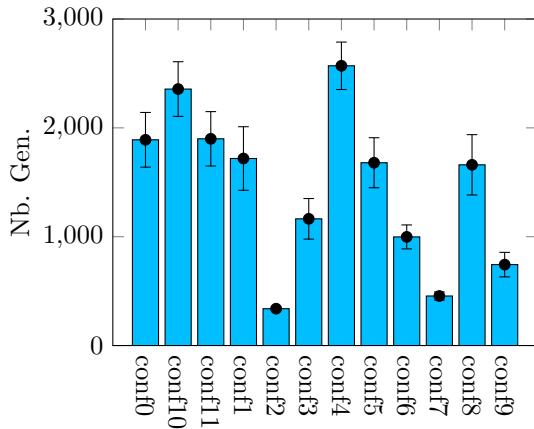


Figure 14: Number of generations before stopping: Contrived sounds

Figure 14 shows the number of generations before reaching the stopping criteria for each target configuration.

Table 5 describes statistics about the proportion of module types in the population over the various

	Prop. choice		Take over gen		Accuracy
	C	NC	C	NC	C
Engine	0.90	0.74	139	129	0.80
FX	0.42	0.44	322	270	0.18
LFO	0.43	0.45	240	317	0.22
Key	0.77	0.38	122	265	0.52
Octave	0.91	0.57	109	174	0.62
Note	0.74	0.38	129	273	0.69

Table 5: Statistics about modules types. C: Contrived sounds, NC: Non-contrived sounds

generations. *Prop. choice* is the proportion of runs where one type was totally taking over in the population. *Accuracy* is the proportion of runs choosing the correct type when one type was taking over in the population. *Take over gen* is the generation as from one type was taking over. The OP-1 has twenty four keys on its keyboard, and it is possible to change the octave from -4 to 4. There is an overlapping of twelve keys between two consecutive values of an octave. It is then possible to produce the same note using two different combinations of octave and key. The last line of Table 5 takes into account this particularity and considers the selected note rather than octave and key taken separately.

Our results suggest that *PresetGen* performs well at finding the right engine type (90 % prop. choice; 80 % accurate) and the right note (74 % prop. choice; 69% accurate). However, we have lower accuracy for the LFO (43 % prop. choice; 22 % accurate) and FX type (42 % prop. choice; 18 % accurate) than the engine type. A possible interpretation of these results is that the engine type and the note have a greater influence on the output sound than the LFO or FX type. The LFO and FX type do not change the nature of the output sound but only alter it. Then, it is more challenging to determine the right type for the FX and LFO.

The correlation between the Euclidian / Hamming distances of individuals to the optimal solution and the objective fitnesses are a good indicator of the problem difficulty (Jones, T. and Forrest, S., 1995). We computed the mean and standard deviation of these distances and the 3 objective fitnesses for each generation. We calculated a *global correlation coefficient* between the average Euclidian/Hamming distances of individuals to the optimal solution and average of each objective fitnesses. Besides we computed a *local correlation coefficient* between the average Euclidian/Hamming distances of individuals to the optimal solution and each average objective fitnesses. Table 6 shows these correlation coefficients. The global correlation coefficients were very high for the Euclidian and Hamming distance. This result explains the tendency of the GA to converge quickly to a punctuated equilibrium of low fitness for the 3 objectives. This tendency is also an illustration of

		FFT	Env	STFT
		Mean(SD)	Mean(SD)	Mean(SD)
Euclidian	Global	0.35(0.46)	0.37(0.45)	0.34(0.41)
	Local	0.04(0.40)	0.09(0.43)	0.09(0.45)
Hamming	Global	0.54(0.34)	0.57(0.27)	0.59(0.31)
	Local	0.04(0.4)	0.006(0.42)	0.06(0.38)

Table 6: Mean and SD for the correlation coefficients

the algorithm ability to converge toward sounds perceptually similar to the target sound. The low local correlation coefficients showed that, once a promising location of the fitness landscape was identified, it was challenging to fine tune the input parameters to converge exactly to the target parameters. One explanation is that the non-deterministic nature of the synthesizer, that causes noise in the evaluation, makes a perfect tuning impossible. Some knobs also have different sensitivity; therefore, a change in one knob can either entail a large change in the output or no change at all.

We call module combination, an OP-1 preset without the knob parameters. For example, {FM synthesis engine, FX delay, LFO element, key 22} is an example of module combinations. The number of distinct module combinations was very low in the Pareto front ($\mu = 3.0$, $SD = 0.2$ over 10 080 possible combinations). These findings suggested that the GA successfully identified a limited number of promising locations in the parameter space that dominate all others. We also notice that, as wanted, each cluster contains only one Engine/LFO/FX combination. A Pareto front affords more flexibility to the user who receives a set of similar sounds rather than a single sound with a simple GA. The user can then make the final choice. In the case of non-contrived sounds, the user can decide what type of perceptual error is more acceptable as interesting.

PresetGen approximates very well the amplitude envelope of the target sound as shown by the very low errors for the envelope objective ($\mu = 0.20$, $SD = 0.02$). Figure 15 shows the error over the best possible FFT and STFT objectives values. As measured by their respective spectral flux, *conf0*, *conf2*, *conf3*, *conf5*, *conf7* and *conf10* are the configura-

tions generating dynamic spectra. The other configurations are generating stationary spectra. We see that the performances of the GA are not significantly better for the target sounds with stationary spectra ($p = 0.07$) than for the target sounds with dynamic spectra ($p = 0.08$). However, we can still observe differences in the GA performances for different groups of target sounds. A first group with (*conf2*, *conf7*, *conf9*) contains the OP-1 target configurations that are the most non-deterministic. Indeed, in this non-determinist context, the best possible objective fitness values are very difficult to determine precisely. Then, it is possible that the GA finds an OP-1 presets outperforming the best possible objective fitness values. This induces a bias when comparing the performances of our system for target sound with stationary spectrum and with a dynamic spectrum. A second group (*conf4*, *conf6*, *conf8*, *conf10* and *conf11*) contains mostly target sounds with stationary spectrum with the exception of *conf10*. *PresetGen* is performing the best for this group as indicated by a very small error compared to the other groups. *Conf10* presents a STFT under the form of a sawtooth wave over time. This common shape doesn't seem to be difficult to approximate for our system even if the spectrum is dynamic. The last group (*conf0*, *conf1*, *conf3*, *conf5*) contains mostly target sounds with dynamic spectra at the exception of *conf1*. The error for this group is the highest. The spectral energy of *conf1* is mainly concentrated in a narrow frequency band. Our system seems to fall into a local minima for non-periodic time-changing spectrum.

8.2.4 Non-contrived sounds

It is more challenging to evaluate the results of the non-contrived sounds trials because we do not have any target parameters or parameter distances to the optimal solution. Even if *PresetGen* has shown good average performances for contrived sounds, it does not necessarily show that it would perform well with any non-contrived sounds. Indeed, the structure and complexity of the fitness landscape depends largely on the chosen target sound. In addition, even if *PresetGen* finds the optimal solution in the synthesizer's search space, we can expect a high error if a non-

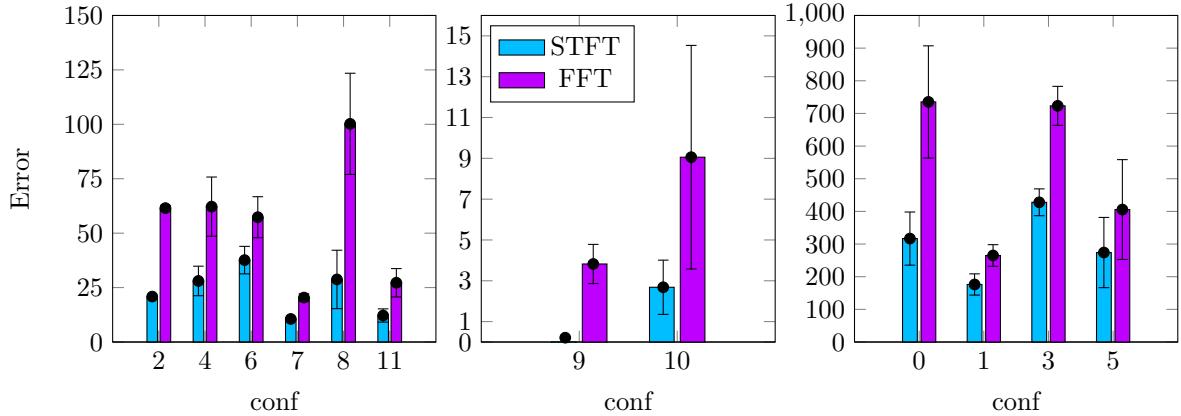


Figure 15: Error: FFT and STFT - Contrived sounds

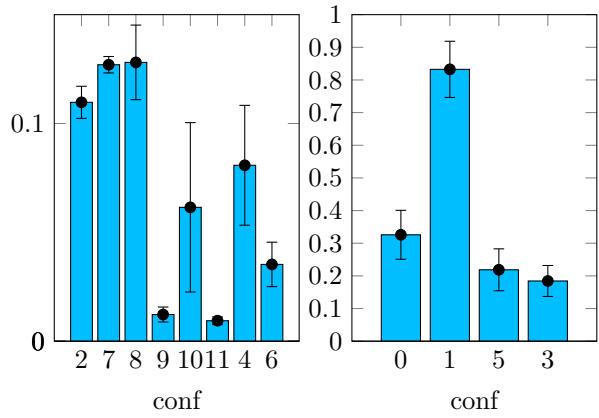


Figure 16: Error: Envelope - Contrived sounds

contrived target sound is actually not reproducible with the synthesizer's architecture.

The mean of $nbGen$, the number of generations before reaching the stopping criteria, was significantly larger for the contrived sounds ($\mu = 1431$, $SD = 86$) than for the non-contrived sounds ($\mu = 1070$, $SD = 50$); $p < 0.001$. This results may seem surprising as the resynthesis problem for a non-contrived sound is more complex than for a contrived sound. However, we designed our algorithm to stop if the cumulative fitness improvement is almost null for each objective (see Section 6.5). *PresetGen* reached faster for the

non-contrived sounds than for the contrived sounds, a level of fitness where improvements were more difficult to obtain, because the potential improvements were more limited in the first case than in the second.

Table 5 describes statistics about the proportion of module types in the population through the generations. As with contrived sounds, a single engine was quickly taking over. However, contrary to the trials with contrived sounds, no key was clearly taking over (Prop choice 38 % against 77 % for contrived sounds); because most of the non-contrived sounds do not have a clearly identified stationary fundamental frequency (*cat meow*, DX-7 and Moog synthesizer sounds with pitch modulation). FX and LFO types were, as with contrived sounds, still challenging to set for the GA (FX prop. choice 44 %, LFO prop choice 45 %).

Figure 17 and Figure 18 respectively show the errors for the FFT/STFT and the Envelope. First, we could hear that the Pareto front sounds were perceptually similar to the targets, which is of importance for real world applications. In terms of number of

	FFT		Env.		STFT	
	Mean	SD	Mean	SD	Mean	SD
C	130.34	17.15	0.20	0.02	248.92	31.24
NC	3163.5	242.67	8.66	0.78	4299.5	262.62

Table 7: Objective best fitness values. C: Contrived sounds, NC: Non-contrived sounds

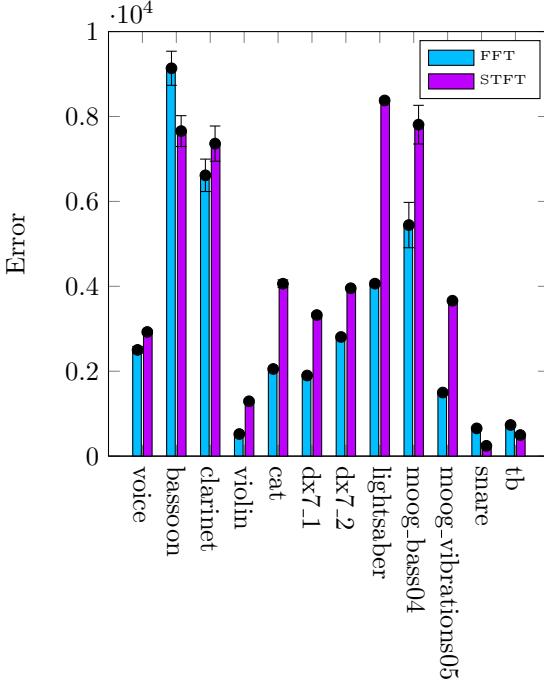


Figure 17: Objective fitnesses: FFT and STFT - Non-Contrived sounds

type combinations, the Pareto fronts were also significantly more diverse ($\mu = 4.3$, $SD = 0.3$) than the ones we got using contrived target sounds ($\mu = 3.0$, $SD = 0.2$); $p < 0.001$. They were also significantly more populated ($\mu = 306$, $SD = 11$; $\mu = 83$, $SD = 21$); $p < 0.001$. These differences caused by the larger problem complexity with the non-contrived sounds. With the concept of clustered Pareto front, the user receives a set of OP-1 presets that produces sounds perceptually similar to the target sound. These OP-1 presets do not involve automatically the use of the same engine, LFO or FX, which gives the users several alternatives of variable quality to approximate a given target sound.

The objective best values for the three objectives, shown in Table 7, were, as expected, significantly worse for the non-contrived sounds than for the contrived sounds ($p < 0.001$, $p < 0.001$, $p < 0.001$).

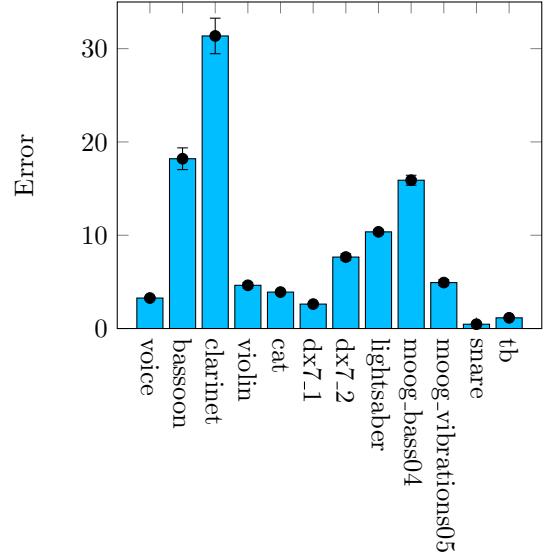


Figure 18: Objective fitnesses: Envelope - Non-Contrived sounds

9 Empirical evaluation with non-contrived sounds

We also conducted an experiment comparing the performance between a (human) sound designer and *PresetGen* on the task of OP-1 preset generation for non-contrived sounds. Then, we expanded our study to three sound designers using the same non-contrived target sounds. We compared their preset generation performance and *PresetGen*'s preset generation performance using the *PresetGen*'s fitness function.

There are two automatic preset generation studies with an empirical evaluation experiment. Mitchell (2010) conducted an empirical evaluation of his FM Modulation Audio Synthesis Parameter Optimization. His evaluation consisted of two listening tests. In the first listening test, he conducted an experiment to research the correlation between fitness error ranking and human perception ranking, thus addressing the - still open - problem of sound similarity. However, the ordinal similarity used in this test has several disadvantages. First, it cannot provide any information if two replications have an insignificant

difference in terms of similarity to the target sound. Participants have to decide that one matching sound is better than others. Second, ordinal similarity evaluation cannot provide a metric of similarity. In addition, we can not compare the similarity measure of different target sounds' matching sounds. In the second listening test, he conducted a qualitative research with five musical instruments sounds. Participants commented on matching sounds in terms of pitch, amplitude envelope, and timbre.

Yee-King, M. and Roth, M. (2008) also evaluated their system, *Synthbot* with expert human users. The application included two VSTi synthesizers: the mdaDX10, a single modulator synthesizer with sixteen parameters, and the mdaJX10, a substrative synthesizer with one noise oscillator and forty parameters. In the first part of the experiment, ten expert users implemented two sounds, one real sound and one synthesized sound, for each of these two synthesizers. They reported the participants' comments on each matching sound.

In our experiment, first, we compared *PresetGen* with one human sound designer on the task of OP-1 preset generation for non-contrived sounds. We researched the relationship between *PresetGen*'s fitness function results and human listener evaluation. Experiment participants evaluated the similarity between the target sounds and matching sounds designed by the (human) designer and *PresetGen*. Hence, we conducted a similarity comparison between a designer and *PresetGen* on the task of OP-1 preset generation for non-contrived sounds. Second, we expanded our experiment to three sound designers to generalize that *PresetGen* is human competitive.

We limited the number of target sounds to eight. The aim was to have an experiment that lasted no longer than fifteen minutes of listening evaluation. We covered diversity on three objectives of *PresetGen*, SFFT, Envelope and FFT; by selecting one sample of each of the following acoustic sounds; *ak47*, *chicken clucking*, *bass guitar*, *trumpet*, *gyil*⁴, *construction noise*, *knife sharpener* and *frog*. These were restricted to have a stationary fundamental frequency

⁴A type of Xylophone from Ocenia

and an applicable envelope considering the capabilities of OP-1.

9.1 Method of the Experiment

9.1.1 Design

We used repeated-measure design in which every participant was exposed to all conditions. The dependent variable is *participant similarity rating* whereas independent variables are,

1. *sound designer* with two-factor levels (*human sound designer* and *PresetGen*),
2. *target sounds* (eight different sound samples),
3. *sound attributes* (*general, pitch, envelope and timbre*).

We also conducted four paired t-test analysis for each sound attributes of each target sound, thirty-two paired t-test in total. In this test, the dependent variable is participant similarity rating whereas the independent variable is a category including two factor levels of *the designer* and *PresetGen*.

9.1.2 Participants

Fourteen participants were recruited from the *IAT 380 Sound Design* class in School of Interactive Arts and Technology at Simon Fraser University so that they had basic experience and knowledge of sound design, sound synthesis and related terms such as timbre, pitch, and envelope.

One participant didn't provide the demographics and experience related questions' answers. Eight participants were female whereas five participants were male. 71% of participants had less than six months of sound design experience whereas 21% of participants had six months to one-year experience in sound design. 29% of participants had no musicianship experience whereas 21% of participants had less than six months of musicianship experience. 14% of participants had one to three years of musicianship experience. 7% of participants had five to ten years of musicianship experience while 14% of participants had

three to five years of musicianship experience. 7% of participants had more than ten years of musicianship experience.

The experiment took ten minutes at most. It was carried out in one session, at the beginning of the course’s weekly workshop. Participants used full range circumaural monitoring headphones and the same type of computers with *M-Audio Fast Track Pro* interfaces. The experiment implemented with an online survey created with *fluidsurveys*⁵. Hence, a browser and simple mouse and keyboard used as the user interface. The actual interface pictured in Figure 19.

9.1.3 Procedure

PresetGen generated the parameters that could approximate each target sound with OP-1. We present the detailed analysis of these runs online (Tatar, K. and Macret, M. and Pasquier, P., 2015). Also, a (human) sound designer, i.e. user, tried to generate the same target sounds with the OP-1 and the matching sounds were recorded. The designer was twenty-six years old male with five years of sound design experience and ten years of musicianship experience.

The design of the experiment involved participants evaluating one target sound and one matching sound at a time because we needed participants to evaluate how similar a matching sound is to its target sound. For each target sound, there were two matching sounds to be evaluated, one generated by the designer and one generated by *PresetGen*. The presentation order was random and balanced within the participant population. Participants could listen to the sounds as much as they wanted. We did not provide any information on how the sounds generated, to prevent any bias against computational creativity (Moffat & Kelly, 2006). Participants answered four questions for each replication. The questions asked general similarity, similarity in terms of pitch, similarity in terms of timbre and similarity in terms of envelope; respectively. Participants answered these

questions given on a 100-point scale in which 100 means the most perceptually similar and 0 means the most perceptual dissimilar.

We used a similarity scale instead of ranking the matching sounds with their similarity to the target sound, to overcome the disadvantages related to the ordinal evaluation mentioned in Section 9. Analysis of a cardinal evaluation provides comparison between matching sounds of different target sounds. Cardinal evaluation shows if the difference between two matching sounds is insignificant or not in terms of similarity to the target sound.

9.2 Results

9.2.1 Quality Comparison

We compared the average participant similarity ratings of *PresetGen*’s and designer’s matching sounds in terms of general similarity, pitch, envelope, and timbre. We conducted a repeated-measures ANOVA⁶ with three independent variables *audio attributes*, *user/PresetGen*, *target sounds* and one dependent variable *similarity to the target sound*. We analyzed tests of within-participants effects for each independent variable and combinations of two independent variables.

The analysis shows that participants rated *PresetGen*’s matching sounds as more similar to the target sound than the (human) designer ones with a mean difference of 16.64 ($F(1, 13) = 48.54, p < 0.001, w^2 = 0.789$)⁷. Analyzing sound attribute category tests within-participant effects, participants rated *PresetGen*’s matching sounds significantly higher than the designer’s matching sounds for all sound attribute categories ($F(3, 39) = 3.62, p = 0.021, w^2 = 0.218$) as shown in Figure 20. Besides, for this independent variable, the Mauchly’s test of sphericity showed that sphericity has not been violated ($p = 0.630$). Spheric-

⁶Analysis of Variances

⁷ w^2 is the effect size and represents what percentage of the variance on the data can be explained by the independent variable variance.

⁵<http://fluidsurveys.com/s/syntheval/>

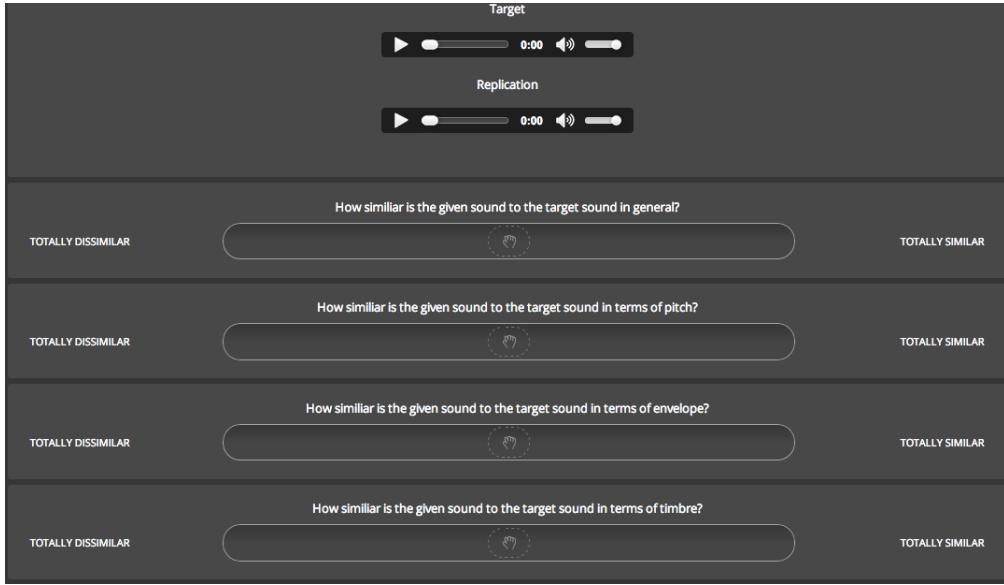


Figure 19: User interface used to evaluate each matching sound on different sound attributes in terms of similarity to the target sound.

ity is defined as ‘the homogeneity of the covariance between pairs of conditions’ (Hinton, 2004).

We have expanded the analysis by examining each sound. Paired t-tests, that is mentioned in Section 9.1.1, shows that the difference in the average similarity rating between *PresetGen* and the designer is insignificant for matching sounds of target sounds; *gyil*, *ak47* and *bass*, for all sound attribute categories. The common property of these sounds is that they have an impulse or a short burst, and they have percussive characteristics. Figure 21 illustrates the average similarity ratings of each matching sound. Paired t-test’s significance values are illustrated online (Tatar, K. and Macret, M. and Pasquier, P., 2015).

9.2.2 The Time Complexity

Although *PresetGen*’s matching sounds are rated more similar in quality, the designer was faster than *PresetGen*. The designer could match three sounds in an hour with OP-1 whereas *PresetGen* generated parameters to match a two-second target sound in five

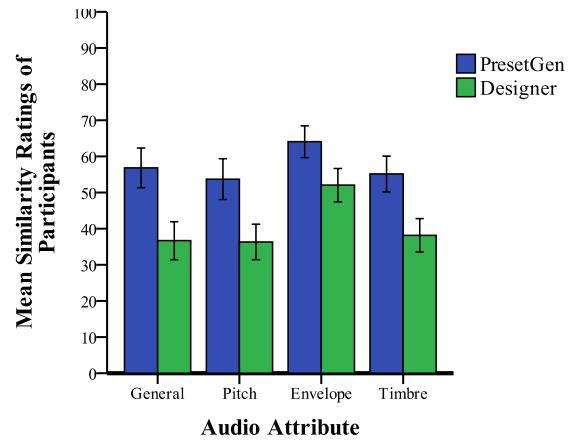


Figure 20: *PresetGen*’s and the designer’s matching sounds’ average similarity ratings by audio attributes, error bars represent 95% confidence interval.

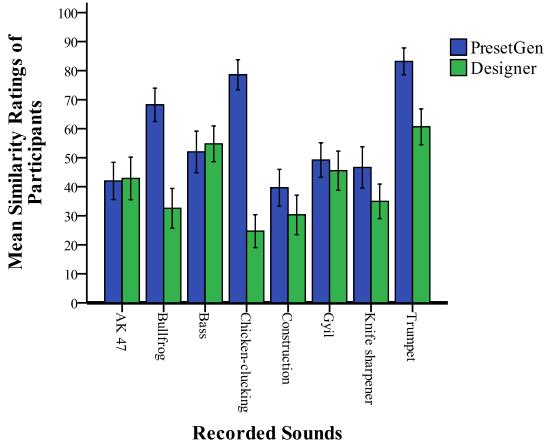


Figure 21: *PresetGen*’s and the designer’s average similarity ratings for each target sound, error bars represent 95% confidence interval.

hours with evolving population of five hundreds individuals over thousands of generations on the Westgrid’s Bugaboo computing cluster with fifty cores. We presented details of our implementation in Section 6.8. It is also important to note that in the case of a sound designer, the designer chooses to stop whereas *PresetGen* stops if one of the stopping criteria is achieved. Moreover, *PresetGen* can be improved with optimization and advancements in the computational processing speed. We explain our future work in Section 10.

9.3 Expansion of empirical evaluation experiment to multiple sound designers

One can argue that empirical evaluation with non-contrived sounds included only one (human) sound designer and results were dependent on the designer’s capabilities. Therefore, we expanded our empirical evaluation experiment with a fitness value comparison of three (human) designers’ and *PresetGen*’s matching sounds. We asked two more sound designers to match same eight target sounds that we used in our empirical evaluation experiment, using

OP-1. We compared matching sounds of designers and *PresetGen* with target sounds using *PresetGen*’s fitness function. Figure 22 shows comparisons of fitness values in three objectives that we used in our multi-objective implementation; *MFCC*, *Envelope* and *SFFT*. Figure 22(a), 22(b) and 22(c) show that *Designer 2* did slightly better than *PresetGen* to match the target sound *construction* in all three objectives. Also, *Designer 3* match the target sound *gyil* better than *PresetGen* in *Envelope* and *MFCC* fitness objectives. Furthermore, Other than these exceptions, *PresetGen*’s matching sounds gave lower fitness values than (human) sound designers’ ones. Therefore, *PresetGen*’s matching sounds were closer to the target sounds than (human) sound designers’ ones, in all objectives with the exceptions that we mention above.

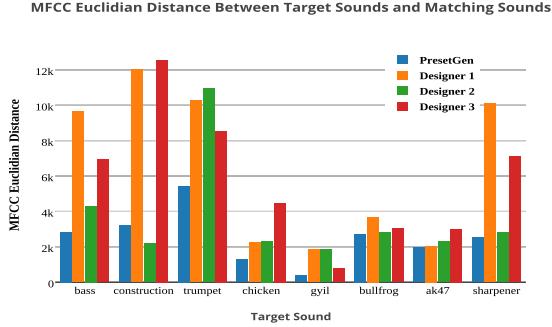
10 Conclusions and Future Work

We focused on the application of evolutionary computation to automatize the task of tuning the parameters of the OP-1, a complex commercial synthesizer developed by Teenage Engineering, to replicate or approximate given target sounds.

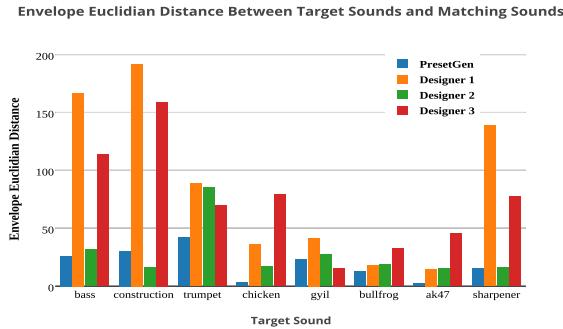
10.1 Contributions

This work provides several contributions to the field of the preset generation for a synthesizer.

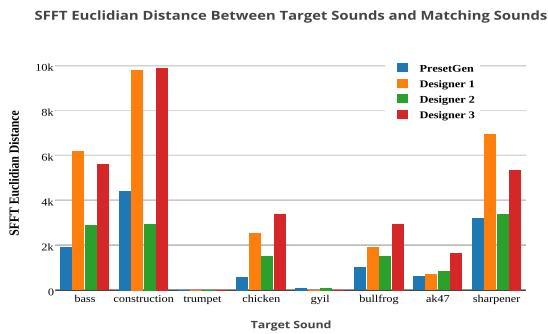
- In Section 6, a Non-dominated Sorting Genetic Algorithm-II (NSGA-II) is presented which incorporates a three objective fitness function, Gray code encoding and a modified crossover operator to preserve population diversity and enable the users to receive a set of solutions rather than a unique solution as with previous systems.
- In Section 6.3, a three objectives fitness function including FFT, Envelope and STFT is developed which addresses some of the difficulties associated with the exploration of a multi-modal search space such as the OP-1 parameters space.



(a) MFCC Euclidian distance between matching sounds and target sounds



(b) Envelope Euclidian distance between matching sounds and target sounds



(c) SFFT Euclidian distance between matching sounds and target sounds

Figure 22: Fitness function objectives comparison of three sound designers and *PresetGen*

- In Section 6.6, a clustering method has been developed to better analyze and explore the set of final solutions. This method is based on k-mean clustering and the silhouette methodology to set the clustering size.

- In Section 8.2, an evaluation is proposed using contrived and non-contrived sounds. Trials revealed the capabilities of *PresetGen* to optimize the parameters of the OP-1 synthesizer to approximate both kind of target sounds.

- In Section 9, we provide an empirical perceptual study that also validates the human competitive nature of *PresetGen*. Instead of ordinal similarity, we use cardinal similarity in the experiment design.

This applied work contributes to the field of sound synthesis using an evolutionary system to find OP-1 synthesizer presets to reproduce given target sounds. Our evaluation, especially the one using contrived target sounds, will make possible to easily compare the performances of *PresetGen* to the performances of future systems developed for the same purpose. In the evaluation with contrived target sounds in Section 8.2, we define the notion of global fitness/distance correlation and local fitness/distance correlation. The high global fitness/distance correlation shows that our algorithm converges, in average, to the region in the preset space where the target preset is located and the low local fitness/distance correlation explains why *PresetGen* is not able to converge exactly to the target preset at the end of the optimization.

The GA system described in this work also contributes to the field of applications in evolutionary computation. *PresetGen* is based on the NSGA-II, a multi-objective genetic algorithm. This multi-objective approach, one that is not common in the field of synthesizer preset generation, has been shown to produce particularly robust results when used to find OP-1 presets to match given target sounds. Using three objectives (FFT, envelope and STFT) instead of only one in previous works, made possible to better solve the complex, multimodal and multi-dimensional optimization problem raised by the real

world synthesizers such as OP-1. These three objectives combined with the intrinsic mechanisms of the NSGA-II (Non-domination sorting, diversity preservation and elitism) made possible to preserve multiple solutions located in diverse regions of the search space and therefore to avoid a premature convergence. A modified crossover operator that prevents the recombination of two individuals with the same genotype, has also been introduced to preserve the diversity. This multi-objective approach also enables users to receive a set of solutions (that can use different synthesis engines, LFO or FX) rather than a unique solution as with previous systems.

PresetGen can be used with other complex commercial synthesizers without any changes except, of course, the genotype encoding of the parameters and the synthesizer simulator.

This work presents an exploration of evolutionary computation applied to synthesizer preset generation with the OP-1 synthesizer.

10.2 Improvements to the system

At present, the evolution needs a large amount of computational power to determine good presets to match a given target sound. We currently evolve population of five hundred individuals over thousands of generations. A single run requires approximatively five hours on a supercomputer with our algorithm distributed on fifty cores. However, as execution time was not a priority in this work, there are numerous optimizations that can be done to the system. For example, the population size and other GA parameters such as mutation and crossover probabilities or the stopping criteria could be adjusted. Another idea would be to reduce the time complexity of extracting the three objective fitness values for every individual of the population. For instance, an optimized temporal segmentation of the STFT could reduce the time computation but also give a more suitable measure for this objective.

FX and LFO module types were also challenging to identify for *PresetGen*. Adding another objective could be a good idea to take into account the nature of these modules. It would also be interesting to separate the optimization of the knob parameters

from the type parameters. In this perspective, using a co-evolution genetic algorithm, where one population representing the types is co-evolved with an other population representing the knob parameters, seems promising.

Results of empirical evaluation with non-contrived sounds pointed out that *PresetGen* is human-competitive on the task of matching a non-contrived sound with impulse characteristics. We also plan to improve *PresetGen* to achieve human-competitiveness with these sounds by updating system's fitness function and experimenting on different objectives.

The sound designer participated in the experiment commented that the synthesizer's user interface provided convenience to match target sounds. For example, to generate a plucked instrument sound, he searched the parameters starting from the presets that had names involving plucked instruments. For this reason, we also research the relationship between the designer's performance and the synthesizer's user interface as a future work.

10.3 Improvements to the evaluation

Our evaluation identified discrepancies in performances using target sounds of different natures. A more in-depth study might explain these differences and would allow us to improve *PresetGen*. For example, target sounds with a dynamic spectrum are more difficult to match than a stationary spectrum. Adding an objective related to this characteristic of the sound could improve the overall system performances. Our target sounds were limited to two seconds duration. Increasing their lengths in a new set of trials would be interesting to study the performance of *PresetGen* for longer sounds.

It was also challenging to compare the performances of *PresetGen* to other systems in the literature. Indeed, different target sounds and performance indicators are used in the different previous works. Developing a benchmark including target sounds of different natures and performances indica-

tors could make possible to easily compare similar systems but also to build on previous works.

Even if our experimentations during the designing phase of our system had lead us to switch from the canonical GA to a multi-objective GA (NSGA-II), it has not been proved formally that the NSGA-II out-performs the canonical GA. In this perspective, it would be interesting to do a direct comparison experiment to compare formally the performance of our NSGA-II system to a canonical GA.

As a future work, we also plan to expand our empirical evaluation experiment to multiple sound designers. Our initial fitness value comparisons with multiple sound designers in Section 9.3 gave encouraging results showing that *PresetGen* match the target sounds better than (human) sound designers in general.

10.4 Applications

A practical application of *PresetGen* would be an online platform in which a user could upload target sounds. The presets search would be evolved offline using *PresetGen* and the resulting OP-1 presets would be sent back to the user by email.

An adapted version of our GA system could also be integrated in an online OP-1 patch randomizer⁸. The idea here would be to use an interactive GA instead of a NSGA-II. The user would be asked to rank by preference the individuals in the population. These rankings would be used to select the individuals for mutation and crossover. It would also be possible to use our NSGA-II system for *background evolution* (McDermott, J. and Griffith, N. and O'Neill, M., 2007). Here, a target sound would be loaded before any user interaction takes place. Our NSGA-II algorithm would then run in the background, attempting to match the target sound. Meanwhile, the user would interact with the system using a GUI in the *foreground*. For each generation, the individuals in the Pareto Front would migrate from *background evolution* to the *foreground evolution*.

⁸<http://op-rand1.appspot.com/welcome.jsf>

PresetGen can also be useful as an educational tool in sound design. The system can point out alternative ways to match same target sound on the synthesizer. Thereby, it can show several different ways to match same target sound.

Working on the OP-1 optimization problem gave us numerous insights on how to solve the harder PureData (PD) patches optimization problem (Macret & Pasquier, 2014). Although this problem presents a lot of similarities to the OP-1 problem, it is more complicated in the sense that PD's audio synthesis architecture is non-linear and modular. With the OP-1, a limited number of synthesis engines, LFO and FX are accessible. On the other hand, PD's building blocks are fundamental synthesis components, such as oscillators or filters. It is virtually possible to generate any synthesis engines, LFO and FX using PD. It makes the search space for the synthesis architecture subsequently more complex for PD than for the OP-1. The number of input parameters is fixed in the OP-1 case, but it can vary in the PD case, making the search even more complex. Given the complexity difference, using the same optimization system for PD than for the OP-1 does not look promising. Instead, we used the idea of using co-evolution to separate the optimization of the synthesis architecture from the synthesis parameters with our Automatic PureData patch generation system. Limiting the number of inputs parameters for PD makes sense in a usability perspective. It also scales well with new promising Evolutionary techniques such as Cartesian Genetic Programming that can evolve graphs: the natural representation for PD patches.

This work brings us closer to making synthesizers more accessible to novice practitioners and help free the musician or composer from tedious calibrations, so that they can focus on producing meaningful sounds and music.

References

- Bozkurt, B. and Yüksel, K. (2011). Parallel evolutionary optimization of digital sound synthesis parameters. In *Proceedings of the conference on*

- applications of evolutionary computation* (pp. 194–203).
- Chan, S., Yuen, J., & Horner, A. (1996). Discrete summation synthesis and hybrid sampling-wavetable synthesis of acoustic instruments with genetic algorithms. In *Proc. of the international computer music conference (icmc)* (p. 49-51).
- Chaudhari, M. and VDharaskar, R. and Thakare, V. (2010). Computing the most significant solution from Pareto front obtained in multi-objective evolutionary. *International Journal of Advanced Computer Science and Applications*, 63-68.
- Deb, K. and Pratap, A. and Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Evolutionary Computation*, 6(2), 182–197.
- Fortin, F. and De Rainville, F. and Gardner, M. and Parizeau, M. and Gagné, C. (2012). DEAP: Evolutionary algorithms made easy. *Journal of Machine Learning Research*, 2171–2175.
- Hinton, P. R. (2004). *SPSS explained*. London: Routledge.
- Horner, A. and Beauchamp, J. (1996). Piecewise-linear approximation of additive synthesis envelopes: a comparison of various methods. *Computer Music Journal*, 20(2), 72–95.
- Horner A. and Beauchamp J. and Haken L. (1993). Machine tongues XVI: Genetic algorithms and their application to FM matching synthesis. *Computer Music Journal*, 17(4), 17–29.
- Jin, Y. and Branke, J. (2005). Evolutionary optimization in uncertain environments-a survey. *Journal on Evolutionary Computation*, 9(3), 303–317.
- Johnson, D. (2002). A theoreticians guide to the experimental analysis of algorithms. *Data structures, near neighbor searches, and methodology: 5th and 6th dimacs implementation challenges*, 59, 215–250.
- Jones, T. and Forrest, S. (1995). Fitness distance correlation as a measure of problem difficulty for genetic algorithms. In *Proceedings of the international conference on genetic algorithms* (pp. 184–192).
- Lai, Y. and Jeng, S.K. and Liu, D.T. and Liu, Y.C. (2006). Automated optimization of parameters for FM sound synthesis with genetic algorithms. In *International workshop on computer music and audio technology*.
- Macret, M., & Pasquier, P. (2014). Automatic Design of Sound Synthesizers As Pure Data Patches Using Coevolutionary Mixed-typed Cartesian Genetic Programming. In *Proceedings of the 2014 Conference on Genetic and Evolutionary Computation* (pp. 309–316). New York, NY, USA: ACM. Retrieved 2015-03-23, from <http://doi.acm.org.proxy.lib.sfu.ca/10.1145/2576768.2598303> doi: 10.1145/2576768.2598303
- Macret, M. and Pasquier, P. and Smyth, T. (2012). Automatic Calibration of Modified FM Synthesis to Harmonic Sounds using Genetic Algorithms. In *Proceedings of sound and music computing conference* (p. 387-394).
- Mathieu, B. and Essid, S. and Fillon, T. and Prado, J. and Richard, G. (2010). Yaafe, an easy to use and efficient audio feature extraction software. In *Proceedings of the International Society for Music Information Retrieval*.
- MATLAB. (2011). *version 7.12.0 (r2011a)*. Natick, Massachusetts: The MathWorks Inc.
- McDermott, J. and Griffith, N. and O'Neill, M. (2007). Evolutionary GUIs for sound synthesis. In *International conference on applications of evolutionary computing* (pp. 547–556).
- Mitchell, T. J. (2010). *An exploration of evolutionary computation applied to frequency modulation audio synthesis parameter optimisation* (engd, University of the West of England). Retrieved 2015-02-09, from <http://www.teamaxe.co.uk>
- Mitchell, T. (2012). Automated evolutionary synthesis matching. *Journal on Soft Computing*, 1–14.
- Mitchell, T. and Creasey, D. (2007). Evolutionary sound matching: A test methodology and comparative study. In *International conference on machine learning and applications* (pp. 229–234).

- Moffat, D., & Kelly, M. (2006, aug). An investigation into people's bias against computational creativity in music composition. In *The third joint workshop on computational creativity*. Trento, Italy: Universita di Trento.
- Peeters, G. (2004). *A large set of audio features for sound description (similarity and classification) in the CUIDADO project* (Tech. Rep.). IRCAM.
- Riionheimo, J. and Välimäki, V. (2003). Parameter estimation of a plucked string synthesis model using a genetic algorithm with perceptual fitness calculation. *Journal on Advances in Signal Processing*, 791–805.
- Rocha, M. and Neves, J. (1999). Preventing premature convergence to local optima in genetic algorithms via random offspring generation. *Multiple Approaches to Intelligent Systems*, 127–136.
- Roth, M. and Yee-King, M. (2011). A comparison of parametric optimization techniques for musical instrument tone matching. In *Proceedings of the audio engineering society convention* (pp. 972–980).
- Rousseeuw, P. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 53–65.
- Schatter, G. and Züger, E. and Nitschke, C. (2005). A synaesthetic approach for a synthesizer interface based on genetic algorithms and fuzzy sets. In *Proceedings of the international computer music conference* (pp. 664–667).
- Tatar, K. and Macret, M. and Pasquier, P. (2015). *Experimental results*. <http://metacreation.net/PresetGen/index.html>. (Last accessed: May 2015)
- Vuori, J. and Välimäki, V. (1993). Parameter estimation of non-linear physical models by simulated evolution-application to the flute model. In *Proceedings of the international computer music conference* (pp. 402–402).
- Wakefield, G., & Mrozek, E. (1996). Perceptual matching of low-order models to room transfer functions. In *Proc. of icmc* (pp. 111–113).
- Westgrid - Compute Canada*. (n.d.). <http://www.westgrid.ca/>. (Last accessed: July 2013)
- Yee-King, M. and Roth, M. (2008). Synthbot: An unsupervised software synthesizer programmer. In *Proceedings of the international conference music conference* (pp. 1–6).

Appendix D

Ranking Based Experimental Music Emotion Recognition

JIANYU FAN
KIVANÇ TATAR
MILES THOROGOOD
PHILIPPE PASQUIER

AS PUBLISHED IN THE PROCEEDINGS OF THE 18TH INTERNATIONAL SOCIETY FOR
MUSIC INFORMATION RETRIEVAL CONFERENCE (ISMIR 2017), 2017

RANKING-BASED EMOTION RECOGNITION FOR EXPERIMENTAL MUSIC

Jianyu Fan, Kivanç Tatar, Miles Thorogood, Philippe Pasquier

Simon Fraser University

Vancouver, Canada

jianyuf, ktatar, mthorogo, pasquier@sfu.ca

ABSTRACT

Emotion recognition is an open problem in Affective Computing the field. Music emotion recognition (MER) has challenges including variability of musical content across genres, the cultural background of listeners, reliability of ground truth data, and the modeling human hearing in computational domains. In this study, we focus on experimental music emotion recognition. First, we present a music corpus that contains 100 experimental music clips and 40 music clips from 8 musical genres. The dataset (the music clips and annotations) is publicly available at: <http://metacreation.net/project/emusic/>. Then, we present a crowdsourcing method that we use to collect ground truth via ranking the valence and arousal of music clips. Next, we propose a smoothed RankSVM (SRSVM) method. The evaluation has shown that the SRSVM outperforms four other ranking algorithms. Finally, we analyze the distribution of perceived emotion of experimental music against other genres to demonstrate the difference between genres.

1. INTRODUCTION

The research in MER proposes computational approaches to recognize the emotion of music. The increasing numbers of MER studies in recent years have been focusing on particular musical genres, such as classical music, pop, rock, jazz, and blues [41]. So far, to our knowledge, MER in experimental music has yet to be explored.

The definition and use of the term experimental music have been an ongoing discussion within the last century. John Cage [15] clarifies the action of experimentalism as “the outcome of which is not foreseen”. Demers [17] defined experimental as “anything that has departed significantly from norms of the time...” [p.7] and continues by the two assumptions of “...that experimental music is distinct from and superior to a mainstream-culture industry and that culture and history determine aesthetic experience” [p.139]. Experimental music does not only rely on harmony and melody [6]. Experimental music explores the continuum between rhythm, pitch, and noise; the notion of organized sound; the expansion of temporal field; and the morphologies of sound. In this study, our definition of experimental music encompasses experimental

electronic music such as acousmatic music, electroacoustic music, noise music, soundscape compositions as well as experimental music with acoustic instruments such as free improvisation or improvised music. We also include Contemporary Art practices that use sound as a medium in our definition of experimental music.

There are many applications in which a computational model of MER for experimental music would be beneficial. MER computational models can be used in the system architecture of Musical Metacreation (MuMe) systems for experimental music. MuMe is the partial or complete automation of musical tasks [34]. A variety of MuMe systems apply machine listening. Machine listening is the computational modeling of the human hearing. In that sense, a computational model for MER in experimental music can be useful to design a machine listening algorithm for a MuMe system. Moreover, we can use computational MER models in the analysis of experimental music works. Also, we can design mood enabled recommendation systems for experimental music albums using a MER model for experimental music.

Still, MER has several challenges. First, music perception can be dramatically different if listeners are from different regions of the world and have various unique cultural backgrounds [5,18]. Second, it is difficult for researchers to collect ground truth data to cover a wide range of population that well distributed in different parts of the world [5]. Third, in the previous studies, researchers designed listening tests that asked participants to annotate the music pieces by rating their emotion perception of the music pieces [41,49]. However, the cognitive load of rating emotion is heavy for participants [9]. This causes the low-reliability of the annotations [19,44]. Fourth, the level of participant’s agreement on the emotion of a music clip varies because the perception of music is subjective. Even for one individual, the ratings can change during a day [49]. Fifth, in the case of experimental music emotion recognition, there is no annotated dataset available. The current MIREX MER task is the case of pop music emotion recognition.

To overcome these difficulties, we designed a ranking-based experiment to collect ground truth annotations based on a crowdsourcing method. Crowdsourcing method is to elicit a large amount of data from a large group of people from online communities [8]. Our ground truth annotations were gathered from 823 annotators from 66 countries, which covers diverse cultural backgrounds. Then, to reduce the cognitive load, we used a ranking-based method to ask participants to do pairwise comparisons between experimental music clips. The ranking



© Jianyu Fan, Kivanç Tatar, Miles Thorogood, Philippe Pasquier. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jianyu Fan, Kivanç Tatar, Miles Thorogood, Philippe Pasquier. “Ranking-Based Emotion Recognition for Experimental Music”, 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

based approach only needs relative comparisons instead of absolute ratings. This improves the objectiveness of the ground truth data. We applied the Quicksort algorithm to select comparisons during the data collection stage to reduce the workload (see Section 4.1). Then, we proposed a SRSVM method and compared it with other ranking algorithms. The results show that SRSVM is better than four other ranking algorithms regarding experimental music emotion recognition.

The database, containing the 140 music clips and the annotations, can be freely downloaded at <http://metacreation.net/project/emusic/>. We believe that public release of such a dataset will foster research in the field and benefit MER communities. The main contributions of this paper are thus four-fold:

- We provide a music corpus, EMusic. The corpus includes 100 experimental music clips and 40 mainstream music clips.
- We use a crowdsourcing method to collect the pairwise ranking data for experimental music clips, and share an annotated experimental music dataset.
- We proposed the SRSVM method for experimental music emotion recognition and compared our approach with other ranking algorithms.
- We compared the annotations of experimental music with that of other music genres.

2. RELATED WORKS

The Music Information Research Evaluation eXchange (MIREX) community evaluates systems for Audio Music Mood Classification every year. Studies have been classified into two major categories based on the model of emotion: categorical and dimensional approaches.

2.1 Categorical Approaches in MER

Categorical MER approaches use discrete affect models to estimate emotion. Discrete affect models propose that we can describe all emotions using a set of basic emotions. These basic emotion categories are happiness, sadness, fear, anger and disgust [22, 33], shame, embarrassment, contempt and guilt [3], as well as exuberance, anxious/frantic and contentment [32]. There is still no consensus on the discrete emotion categories of music [32].

In the previous studies with categorical MER approaches, researchers conducted experiments to collect the ground truth annotations. Then, researchers used the audio features of music clips with classification methods to model the relationship between audio features and emotion categories [23, 45, 46].

2.2 Dimensional Approaches in MER

Dimensional affect models use a Cartesian space with continuous dimensions to represent emotions [7, 14, 40, 48]. The simplest dimensional affect model has two dimensions: valence and arousal. Other dimensional affect models with additional dimensional such as tension, potency, and dominance have also been proposed in the literature [32]. MER studies use dimensional affect models to compute continuous values that represent the emotion of audio samples. These studies focus on continuous ma-

chine learning models such as regression models. Researchers gather the ground truth data by conducting an evaluation experiment in which the participants label the emotion music clips on a dimensional affect grid.

2.3 Rating or Ranking

Affective ratings instruments have been used for collecting affective annotations. Researchers have used such tools in video emotion recognition [27, 30], music emotion recognition [11], speech emotion recognition [35], soundscape emotion recognition [20] and movement emotion recognition [43]. However, recent studies show that rating based experiments have limitations and fundamental flaws [13]. Rating-based experiments neglect the existence of interpersonal differences on the rating process. In addition, rating emotion in a continuum is difficult because annotators tend to score the samples based on the previous ratings instead of their non-biased feelings [44]. Yang and Lee indicated that the rating-based approach imposes a heavy cognitive load on the subjects [48]. Moreover, the contextual situation of annotators can affect the consistency of ratings [12].

Ranking has been an alternative approach for eliciting responses from subjects [9, 39, 48]. Metallinou and Narayanan found that there is a higher Inter-annotator reliability when people were asked to describe emotions in relative terms rather than in absolute terms [2]. Yannakakis et al. also showed that the inter-rater agreement of the ordinal data is significantly higher than that of the nominal data [12].

Yang and Chen designed a ranking-based experiment to collect ground truth data and build a ranking model recognize the perceived emotion of pop music [9]. The result showed that the ranking-based approach simplifies the annotation process and enhances the Inter-annotator reliability. Hence, we designed a ranking-based method to for experimental music emotion recognition, where annotators made pairwise comparisons between two audio clips based on valence and arousal.

2.4 Emotion Taxonomy

According to previous studies [1, 24], two types of emotions are at play when listening to music.

- Perceived emotion: Emotions that are communicated by the source.
- Induced emotion: Emotional reaction that the source provokes in listeners.

The perceived emotion is more abstract and objective. It is the emotion the source conveys. The perceived emotion of happy songs is always “happy”. However, the induced emotion is more subjective. The same happy music may not necessarily induce happiness in the listener. In this study, we focus on the perceived emotion of music clips because it is more objective.

3. DATA COLLECTION

To build a MER system for experimental music, we first built an experimental music corpus: EMusic. Then, we collected emotion annotations using a crowdsourcing method.

3.1 Corpus Construction

In EMusic corpus, there are 100 experimental music clips and 40 music clips from 8 musical genres, including blues, classical, country, electronic, folk, jazz, pop and rock. The 100 experiment music clips are extracted from 29 experimental music pieces, which are high quality works of Electroacoustic music. The 40 music clips are selected from 1000 songs database [29]. We segmented these compositions using multi-granular novelty segmentation [31] provided in the MIRTToolbox [32]. Using this automatic segmentation method, we ensure that each segment is consistent. Then, we manually chose novel clips to create a homogeneous and consistent corpus that would not disturb the listeners. A 0.1 seconds fade in/out effect has been added to each audio clip.

Music clips are converted to a format in wav (44100 Hz sampling frequency, 32 bits precision and mono channel). All the audio samples are normalized. Regarding the duration, Xiao et al. [50] showed that the use of six to eight seconds is good for presenting stable mood for classical music segments. Fan et al. [19] indicated that the duration of six seconds is long enough for soundscape emotion recognition. Following the previous study, we aimed for the average duration of 6 seconds in this experiment (Mean: 6.20s, Std: 1.55s). The duration of clips varies because of the automatic segmentation by novelty.

3.2 Select Comparisons

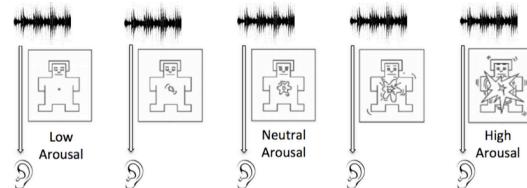
To create a robust set of annotations, we need multiple annotations per pairwise comparison of audio clips. Baveyes et al. [44] found that collecting three annotations per comparison is a good compromise between the cost and the accuracy of the experiment. Therefore, we follow this approach for its feasibility within our experiment.

To efficiently create pairwise comparisons presented to the listeners, we use a Quicksort algorithm [44]. For the first iteration of the algorithm, we select one audio sample as the pivot. All remaining clips are to be compared with the pivot so that the algorithm generates 139 comparisons. We then collect three annotations for each comparison and determine the result to be the one that provided by at least two annotators. In the case that we did not select a pivot that has the lowest or the highest valence or arousal, we end up with two separate sets after the first iteration. Therefore we repeatedly select a new pivot in each set until each audio clip received a rank of valence and a rank of arousal from 1 to 140. The computational complexity of the Quicksort algorithm is $O(N \log N)$.

3.3 Online Experiment

We conduct an online experiment to annotate our corpus of experimental music clips with affective labels. We used the CrowdFlower¹ platform to crowd source annotations from people online. To sort the 140 music clips based on valence and arousal independently, we launched one task for valence and another task for arousal.

See below for some more examples of low and high arousal:



Steps:

- Listen to both audio recordings.
- Determine which audio recording to choose based on the arousal expressed by the audio recordings.
- Mark the corresponding radio button.

Notices:

- We ask you to use circumaural head-phones (headphones that completely surround the ears) to listen to the audio clips.



- Please make sure you are in a quiet environment when you do the experiment.
- Do not start playing several audio files simultaneously.



Which audio has higher arousal?

- left
- right

Figure 1. The interface of crowdsourcing study.

At the beginning of the annotation process, subjects are provided with the terminology of arousal and valence. In our experiment, we used valence to describe perceived

pleasantness of the sound. We provided subjects with the Self-Assessment Manikin [28] at the beginning of the task to make sure the task was understood. The Self-Assessment Manikin is a pictorial system used in experiments to represent emotional valence and arousal axes. Its non-verbal design makes it easy to use regardless of age, educational or cultural background. We modified the pictorial system by adding arrows to inform annotators that we were collecting perceived emotion.

We requested annotators to follow a tutorial to get familiar with the annotation interface. Annotators were notified that they were required to use headphones to listen to the audio clips. We asked them to turn the volume up to a comfortable level given a test signal. Annotators were then presented with a quiz, where 5 gold standard comparisons were provided. These comparisons were easily comparable regarding valence and arousal, which were carefully selected by experts. The annotators could continue to the task only if they achieve an 80% of accuracy in the quiz.

To ensure the quality of the annotations, we tracked annotators' performance by inserting gold standard comparisons throughout the tasks. Similar to the comparisons in the quiz, these 5 comparisons were easily comparable regarding valence and arousal. If their answers were not the same as the default answer, they would be noticed by a pop out window. If they had strong reason to explain their answer, they could message the reason to us. This also affects annotators' reputation on the CrowdFlower.

¹ <https://www.crowdflower.com/>

Annotators could listen repeatedly to an audio clip. After an annotator had listened to both audio clips, the option to enter the response was presented in the form of an input button. For easing the fatigue that increases naturally during manual data annotation [2], they could pause the annotation process at any time and continue at a later stage. The volume control bar was disabled so that annotators could not adjust the individual volumes themselves. An annotator had to rank 5 pairs of clips before being paid US\$0.05 and was able to exit the task at any time.

3.4 Annotation Results

A total of 823 annotators performed the task from 66 different countries. Most of the workers are Venezuelans (31.71%), Brazilian (6.93%), Serbian (6.44%), Russian (5.95%) and Bosnians (5.10%). The annotators were from the world population and it is unlikely they have a background in experimental music. This avoids the potential bias brought by experts.

Each pair was displayed to annotators until three annotations are collected for this pair. 823 annotators provided 2817 comparisons for arousal and 2445 comparisons for valence. The 823 trusted annotators had an average accuracy of 91.81% in the quiz. Annotators took approximately 13s to perform a task. This also proves that annotators carefully listened to both music clips.

Categories	Arousal	Valence
Percent Agreement	0.839	0.801
Krippendorff's α	0.360	0.222

Table 1. Inter-annotator reliability.

We evaluate the Inter-annotator reliability based on percent agreement and Krippendorff's α . Percent agreement calculates the ratio between the number of annotations that are in agreement and the total number of annotations. However, percent agreement overestimates inter-annotator reliability because it does not consider the agreement expected by chance. Krippendorff's α is more flexible and allows missing data (comparisons can be annotated by any number of workers). Thus, no comparisons are discarded to compute this measure. Their values can range from 0 to 1 for Percent agreement and from -1 to 1 for Krippendorff's alpha.

In Table 1, the inter-annotator reliability is similar to other emotion studies [30, 44]. The percent agreement indicates that annotators agreed on 83.9% and 80.1% of comparisons. The value of Krippendorff's α is between 0.21 to 0.40, which indicates a fair level of agreement.

4. LEARN TO RANK

4.1 Standard Ranking Algorithms

The state-of-the-art ranking algorithms can be three categories: the pointwise approach [42], the pairwise approach [36] and the listwise approach [10]. The pointwise approach learns the score of the samples directly. The pointwise approach takes one train sample at a time and trains a classifier/regressor based on the loss of the single sample. The pairwise approach solves the ranking problems by using a pair of samples to train and provides an

optimal ordering for the pair. Listwise methods try to minimize the listwise loss by evaluating the whole ranking list. Each ranking algorithm assigns a ranking score to each sample, and rank the sample based on the score.

In the following, we introduce five ranking algorithms: ListNet, Coordinate Ascent, RankNet, RankBoost and RankSVM. ListNet is a listwise ranking algorithm [10], which uses neural networks to predict the ranking score. The algorithm calculates the probability of the sample ranking within top-k, and computes the difference between the probability distribution of predicted ranks and ground truth data based on cross entropy. Coordinate Ascent algorithm is a gradient-based listwise method for multi-variate optimization [16]. It directly optimizes the mean of the average precision scores for each ranking. RankNet is a pairwise ranking algorithm, which predicts the ranking probability of a pair of samples $\langle A, B \rangle$. If sample A receives a higher ranking score than that of sample B, then the object probability \bar{P}_{AB} equals 1, otherwise, \bar{P}_{AB} equals 0. The loss function of RankNet is the cross-entropy between the predicted probability and the object probability. RankBoost is another pairwise ranking algorithm [47]. It replaces training samples with pairs of samples to learn the association between samples. RankSVM is a common pairwise method extended from support vector machines [36]. The difference between features vectors of a pair of training samples can be transformed to a new feature vector to represent the pair. RankSVM converts a ranking task to a classification task.

4.2 Searching Strategies

Given a test sample, a ranking model provides a ranking score regarding valence/arousal. A ranking score is a real number. To obtain the predicted rank of the test sample based on the ranking score, we used two search strategies: one-by-one search and smoothed binary search.

4.2.1 One-by-One Search

First, we obtain predicted ranking scores of the entire training set and the test sample. Then, we sorted all clips by ranking score to obtain the predicted ranking of the test sample. Ties are unlikely to happen since we set the value of the score retains 6 digits after the decimal point.

4.2.2 Smoothed Binary Search

Smoothed binary search compares the ranking score of a test sample with the ranking scores of pivots selected from the training set to find the rankings of a test sample along the valence/arousal axis. We add a smoothed window to traditional binary by selecting a group of pivots instead of one pivot. Following is the description of the smoothed binary search:

- Given a test sample, pick an odd number of clips from the training set that are consecutive on the valence/arousal axis as pivots. The odd number of clips avoids the ties. The group of pivots has the medium value of valence/arousal among the subset.
- Predict the ranking score for the group of pivots and the test sample, and compare their ranking score. The test sample with a score of less than half of the pivots comes before the pivots, while the test

- sample with a score greater than half of the pivots comes after pivots.
- Recursively apply the above steps until the size of subsets is 2. The average ranking of these two training samples is the predicted rankings.

4.3 SRSVM

We propose the SRSVM for experimental music emotion recognition. The training of SRSVM is the same as standard RankSVM. During the testing/ranking stage, SRSVM finds the predicted ranking of the test sample based on the smoothed binary search.

5. PERFORMANCE ANALYSIS

5.1 Features Selection

We began with a feature set including rms, brightness, loudness, spectral slope, spectral flux, spectral rolloff, attack leap, regularity, pulse clarity, hcdf, inharmonicity, perceptual sharpness, pitch, key, tempo, and 12 MFCCs. We used 23-ms analysis windows and calculated the mean and standard deviation to represents signals as the long-term statistical distribution of local spectral features, which ended up with a 56-dimension feature vector [21]. We used MIRToolbox [32] and YAAFE [4] libraries to extract audio features.

Selected Features
Mean of Root Mean Square
Standard deviation of Root Mean Square
Standard deviation of Brightness
Mean of MFCC 1
Standard deviation of MFCC 2
Standard deviation of MFCC 8
Mean of MFCC 12
Mean of Hcdf
Mean of Loudness
Standard deviation of Loudness
Mean of Regularity

Table 2. Selected features for predicting valence/arousal

Before training the model, we build a feature selector that removes all low-variance features over the entire corpus to select a subset of discriminative features. The threshold of variance is 0.02, which is chosen as a heuristic value. This step kept 43 features out of 56 features. Then, we used a random forests method, which has ten randomized decision trees to evaluate the importance of features based on the Gini impurity index. We ended up having an 11-dimensional feature vector (see Table. 2). Because our dataset includes 100 experimental music clips and 40 clips belong to other genres, we tested the ranking algorithms using the whole dataset and the subset of experiment music separately.

5.2 Comparing with Ranking Algorithms

We evaluate the ranking algorithms of experimental MER using Goodman-Kruskal gamma (G). Goodman-Kruskal gamma measures the association between the predicted rankings and the ground truth annotations [37, 38]. G de-

pends on two measures: the number of pairs of cases ranked in the same order on both variables (number of concordant, N_s) and the number of pairs of cases ranked in reversed order on both variables (number of discordant, N_D). G ignores ties. In our experiment, we had no ties. G is close to 1 indicate strong agreement, -1 for total disagreement, and 0 if the rankings are independent.

$$G = \frac{N_s - N_D}{N_s + N_D} \quad (1)$$

We used the leave-one-out validation method to compare the SRSVM with ListNet, RankNet, Coordinate Ascent, and RankBoost. For a given test sample, ranking algorithms output a predicted valence/arousal score. To obtain the predicted rankings of the whole test set, we used one-by-one searching strategy and smoothed binary search strategy. Then, we measured the gamma between the predicted rankings and the ground truth annotation.

As we can see from Table 3, when we use SRSVM, we obtain the best performance when the windows size is three samples (G : 0.733, $p < 0.001$). When the window size is 1, the test sample will be compared with one pivot iteratively until it falls into a small interval. This becomes a standard binary search. After adding a smoothed window, the test sample is compared with a group of pivots. This increases the accuracy of predicting whether the test sample is larger or smaller than the pivots.

Algorithm	One-by-One Search	Smoothed Binary Search (Number of samples)		
		1	3	5
ListNet	0.044	0.088	0.057	0.022
RankNet	0.096	0.386	0.269	0.255
Coordinate Ascent	0.191	0.436	0.387	0.486
Rank-Boost	0.619	0.679	0.697	0.717
RankSVM	0.398	0.690	0.733 SRSVM	0.697 SRSVM

Table 3. Goodman-Kruskal gamma of ranking algorithms for arousal recognition using the whole dataset

Method	One-by-One Search	Smoothed Binary Search (Number of samples)		
		1	3	5
ListNet	0.015	0.002	0.049	0.002
RankNet	0.063	0.155	0.055	0.260
Coordinate Ascent	0.016	0.130	0.195	0.254
Rank-Boost	0.438	0.467	0.345	0.440
RankSVM	0.333	0.490	0.573 SRSVM	0.556 SRSVM

Table 4. Goodman-Kruskal gamma of ranking algorithms for valence recognition using the whole dataset

When using the whole dataset, the valence recognition is harder than arousal recognition. However, the SRSVM still obtains the best performance ($G: 0.573, p < 0.001$).

Method	One-by-One Search	Smoothed Binary Search (Number of samples)		
		1	3	5
ListNet	0.001	0.037	-0.013	0.013
RankNet	0.110	0.096	0.242	0.299
Coordinate Ascent	0.237	0.515	0.519	0.556
Rank-Boost	0.698	0.741	0.740	0.748
RankSVM	0.300	0.776	0.801 SRSVM	0.776 SRSVM

Table 5. Goodman-Kruskal gamma of ranking algorithms for arousal recognition using the subset that only contains experimental music clips.

As Table 5 shows, when we only consider experimental music, the Gamma statistic of SRSVM for arousal recognition has the best result ($G: 0.801, p < 0.001$). The results of the experimental music case are better than the results of the case including clips of all genres.

Method	One-by-One Search	Smoothed Binary Search (Number of samples)		
		1	3	5
ListNet	0.115	0.037	-0.012	0.036
RankNet	0.058	0.116	0.246	0.277
Coordinate Ascent	0.067	0.100	0.131	0.106
Rank-Boost	0.167	0.236	0.279	0.346
RankSVM	0.434	0.570	0.795 SRSVM	0.628 SRSVM

Table 6. Goodman-Kruskal gamma of ranking algorithms for valence recognition using the subset that only contains experimental music clips.

Table 6 shows that when we only consider experimental music, the Gamma statistic of SRSVM for valence recognition ($G: 0.795, p < 0.001$) is significantly higher than using the whole dataset.

From Table 3-6, we can see the best performing model is SRSVM with 3 samples as the smoothed window. The second best performing model is SRSVM with 5 samples as the smoothed window. This result implies that a good emotion-recognition can be obtained by using SRSVM.

5.3 Comparing between Experimental Music and Other Genres

We convert the rankings to ratings to visualize the distribution of the ranking data. This illustration has two assumptions. First, the distances between two successive rankings are equal. Second, the valence and arousal are in the range of [-1.0, 1.0].

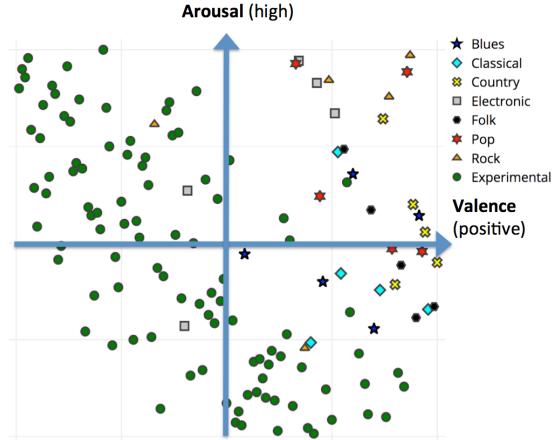


Figure 2. The distribution of the ground truth annotations, the green dots represent experimental music clips

From Figure 2, it can be observed that other genres have both higher perceived valence and arousal comparing to experimental music. Because we have only 5 samples per genre, we need to have a large ground truth dataset to prove that assumption. The figure also shows the negative correlation between valence and arousal of experimental music clips. To test this, we run a Pearson correlation test on the ground truth data. Our Pearson correlation coefficient is -0.3261, which indicates there is a weak negative correlation between the two dimensions.

6. CONCLUSIONS AND FUTURE WORKS

We present an annotated dataset for experimental music emotion recognition. 140 music clips are ranked along the valence and arousal axis through a listening experiment. It is available at <http://metacreation.net/project/emusic/>. We presented a SRSVM method to predict rankings of experimental music clips regarding valence/arousal and compared SRSVM with other ranking method. We also compared the valence and arousal of experimental music with that of the music of other genres, which shows other genres of music have both higher perceived valence and arousal than experimental music.

Even with the smaller number of clips, we found other genres have both higher perceived valence and arousal comparing to experimental music. In the future, we plan to compare the perceived emotion of different genres by collecting a larger dataset.

7. REFERENCES

- [1] A. Kawakami, K. Furukawa, K. Katahira and K. Okanoya, "Sad music induces pleasant emotion," *Front Psychol* Vol. 4, No. 311, 2013.
- [2] A. Metallinou and S. Narayanan, "Annotation and processing of continuous emotional attributes: challenges and opportunities," *IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, pp. 1–8, 2013.

- [3] A. Ortony and T. J. Turner, "What's basic about basic emotions?" *Psychological review*. Vol. 97, No. 3, pp. 315-331, 2014.
- [4] B. Mathieu, S. Essid, T. Fillon, J. Prado, and G. Richard, "Yaafe, an Easy to Use and Efficient Audio Feature Extraction Software," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 441-446. 2010.
- [5] C. J. Stevens, "Music perception and cognition: A review of recent cross-cultural research," *Topics in Cognitive Science*, Vol. 4, No. 4, pp. 653-667, 2012.
- [6] C. Palombini, "Pierre Schaeffer. 1953: Towards an Experimental Music," *Music and Letters*, Vol. 74, No. 4, pp. 542-57, 1993.
- [7] D. Liu, L. Lu, and H.-J. Zhang, "Automatic mood detection from acoustic music data," *Proceedings of the International Symposium Music Information Retrieval*, pp. 81-87, 2003.
- [8] D. McDuff, "Crowdsourcing affective responses for predicting media effectiveness," *Ph.D. Dissertation*. Massachusetts Institute of Technology, 2014.
- [9] D. Yang and W.-S. Lee, "Disambiguating music emotion using software agents," *Proceedings of the International Conference on Music Information Retrieval*, 2004.
- [10] F. Xia, T.-Y. Liu, J. Wang, W.-S. Zhang, and H. Li, "Listwise approach to learning to rank: Theory and algorithm," *Proceedings of the IEEE International Conference on Machine Learning*, pp. 1192-1199, 2008.
- [11] F. Weninger, F. Eyben, and B. Schuller, "On-line continuous-time music mood regression with deep recurrent neural networks," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2014.
- [12] G. N. Yannakakis and H. P. Matínez, "Grounding Truth via Ordinal Annotation," *Proceedings of the International Conference on Affective Computing and Intelligent Interaction*, 2015.
- [13] G. N. Yannakakis, H. P. Martínez, "Ratings are overrated!" *Frontiers on Human-Media Interaction*, Vol. 2, No. 13, 2015.
- [14] J. A. Sloboda and P. N. Juslin, "Psychological perspectives on music and emotion," in *Music and Emotion: Theory and Research*, Oxford University Press, 2001.
- [15] J. Cage, *Silence: Lectures and Writings*, Wesleyan, 1961.
- [16] J. Chen, C. Xiong, and J. Callan, "An empirical study of learning to rank for entity search," *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2016.
- [17] J. Dermers, *Listening through the Noise: The Aesthetics of Experimental Electronic Music*, Oxford University Press, 2010.
- [18] J. Fan and M. Casey, "Study of Chinese and UK hit songs prediction," *Proceedings of the International Symposium on Computer Music Multidisciplinary Research*, pp. 640-652, 2013.
- [19] J. Fan, M. Thorogood, P. Pasquier, "Automatic Soundscape Affect Recognition Using A Dimensional Approach," *Journal of the Audio Engineering Society*, Vol. 64, No. 9, pp. 646-653, 2016.
- [20] J. Fan, M. Thorogood, and P. Pasquier, "Automatic Recognition of Eventfulness and Pleasantness of Soundscape," *Proceedings of the 10th Audio Mostly*, 2015.
- [21] J. J. Aucouturier and B. Defreville, "Sounds like a park: A computational technique to recognize soundscapes holistically, without source identification," *Proceedings of the International Congress on Acoustics*, pp. 621-626, 2009.
- [22] J. Panksepp: *Affective Neuroscience: The Foundation of Human and Animal Emotions*, Oxford University Press, 1998.
- [23] K. Bischoff, C. S. Firat, R. Paiu, W. Nejdl, C. Laurier, and M. Sordo, "Music mood and theme classification—a hybrid approach," *Proceedings of the International Conference on Music Information Retrieval*, pp. 657-662, 2009.
- [24] K. Kallinen and N. Ravaja, N, "Emotion perceived and emotion felt: Same and different," *Musicae Scientiae*, Vol. 5, No. 1, pp. 123-147, 2006.
- [25] K. Svore, L. Vanderwende, and C. Burges, "Enhancing single-document summarization by combining RankNet and third-party sources," *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pp. 448-457, 2007.
- [26] L. A. Goodman and W. H. Kruskal, "Measures of Association for Cross Classifications," *Journal of the American Statistical Association*. Vol. 49, No. 268, pp.732-764, 1954.
- [27] L. Devillers, R. Cowie, J.-C. Martin, E. Douglas-Cowie, S. Abrilian, and M. McRorie, "Real life emotions in French and English TV video clips: an integrated annotation protocol combining continuous and discrete approaches," *Proceedings of the International conference on Language Resources and Evaluation*, 2006.

- [28] M. M. Bradley and P. J. Lang, "Measuring emotion: the self- assessment manikin and the semantic differential," *Journal of behavior therapy and experimental psychiatry*, Vol. 25, No. 1, pp. 49–59, 1994.
- [29] M. Soleymani, M. N. Caro, E. M. Schmidt, C.-Y. Sha, and Y.-H. Yang, "1000 songs for emotional analysis of music," *Proceedings of the 2nd ACM International Workshop on Crowdsourcing for Multimedia*, pp. 1–6, 2013.
- [30] N. Malandrakis, A. Potamianos, G. Evangelopoulos, and A. Zlatintsi, "A supervised approach to movie emotion tracking," *Proceedings of the IEEE International Conference Acoustics, Speech and Signal Processing*, pp. 2376–2379, 2011.
- [31] O. Lartillot, D. Cereghetti, K. Eliard, and D. Grandjean, "A simple, high-yield method for assessing structural novelty," *Proceedings of the 3rd International Conference on Music & Emotion*, 2013.
- [32] O. Lartillot, P. Toivainen, and T. Eerola, "A Matlab Toolbox for Music Information Retrieval," in: C. Preisach, H. Burkhardt, L. Schmidt-Thieme, P. D. R. Decker, (Eds), *Data Analysis, Machine Learning and Applications. Studies in Classification, Data Analysis, and Knowledge Organization*, pp. 261–268, Springer, Berlin, Heidelberg, 2008.
- [33] P. Ekman, "An argument for basic emotions," *Cognition and Emotion*. Vol. 6, No. 3, pp.169–200, 1992.
- [34] P. Pasquier, A. Eigenfeldt, O. Bown, and S. Dubnov, "An Introduction to Musical Metacreation," *ACM Computers In Entertainment, Special Issue: Musical Metacreation*, Vol. 14, No. 2, 2016.
- [35] R. Elbarougy and M. Akagi, "Speech Emotion Recognition System Based on a Dimensional Approach Using a Three-Layered Model," *Proceedings of the Signal & Information Processing Association Annual Summit and Conference*, 2012.
- [36] R. Herbrich, T. Graepel, and K. Obermayer, "Support vector learning for ordinal regression," *Proceedings of International Conference on Artificial Neural Network*, 1999.
- [37] R. Morris, "Crowdsourcing workshop: The emergence of affective crowdsourcing," *Proceedings of the Annual Conference Extended Abstracts on Human Factors in Computing Systems*, 2011.
- [38] R. Morris and D. McDuff, "Crowdsourcing techniques for affective computing," in R.A. Calvo, S.K. DMello, J. Gratch and A. Kappas (Eds). *Handbook of Affective Computing*, Oxford University Press, 2014.
- [39] S. Ovadia, "Ratings and rankings: Reconsidering the structure of values and their measurement," *International Journal of Social Research Methodology*, Vol. 7, No. 5, pp. 403–414, 2004.
- [40] T. Eerola, O. Lartillot, and P. Toivainen, "Prediction of multidimensional emotional ratings in music from audio using multivariate regression models," *Proceedings of the International Symposium Music Information Retrieval*, pp. 621–626, 2009.
- [41] T. Eerola, Tuomas, and J. K. Vuoskoski. "A Review of Music and Emotion Studies: Approaches, Emotion Models, and Stimuli," *Music Perception: An Interdisciplinary Journal*, Vol. 30, No. 3, pp. 307–340, 2013.
- [42] T. Y. Liu, "The Pointwise Approach," in *Learning to rank for information retrieval*. Berlin: Springer-Verlag Berlin and Heidelberg GmbH & Co. K, 2011.
- [43] W. Li, P. Pasquier, "Automatic Affect Classification of Human Motion Capture Sequences in the Valence-Arousal Model," *Proceedings of the International Symposium on Movement and Computing*, 2016.
- [44] Y. Baveye, E. Dellandrea, C. Chamaret, and L. Chen, "LIRIS-ACCEDE: A video database for affective content analysis," *IEEE Transactions on Affective Computing*, Vol. 6, No. 1, pp. 43–55, 2015.
- [45] Y. Feng, Y. Zhuang, and Y. Pan, "Popular music retrieval by detecting mood," *Proceedings of the International Conference on Information Retrieval*, pp. 375–376, 2013.
- [46] Y. E. Kim, E. M. Schmidt, R. Migneco, B. G. Morton, P. Richardson, J. Scott, J. A. Speck, and D. Turnbull, "Music emotion recognition: A state of the art review," *Proceedings of the International Conference on Music Information Retrieval*, 2010.
- [47] Y. Freund, R. Iyer, R. Schapire, and Y. Singer, "An efficient boosting algorithm for combining preferences," *Proceedings of the International Conference on Machine Learning*, pp. 170–178, 1998.
- [48] Y.-H. Yang and H. Chen, "Ranking-based emotion recognition for music organization and retrieval," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 19, No. 4, pp. 762–774, 2011.
- [49] Y.-H. Yang and H.-H. Chen, "Machine recognition of music emotion: A review," *ACM Trans. Intel. Systems & Technology*, Vol. 3, No. 3, 2012.
- [50] Z. Xiao, E. Dellandrea, W. Dou, and L. Chen, "What is the best segment duration for music mood analysis?" *Proceedings of the International Workshop on Content-Based Multimedia Indexing*, pp. 17–24, 2008.

Appendix E

Quantitative Analysis of the Impact of Mixing on Perceived Emotion of Soundscape Recordings

JIANYU FAN
MILES THOROGOOD
KIVANÇ TATAR
PHILIPPE PASQUIER

AS PUBLISHED IN PROCEEDINGS OF SOUND AND MUSIC COMPUTING (SMC 2018),
2018

QUANTITATIVE ANALYSIS OF THE IMPACT OF MIXING ON PERCEIVED EMOTION OF SOUNDSCAPE RECORDINGS

Jianyu Fan

Simon Fraser University

jianyuf@sfu.ca

Miles Thorogood

Simon Fraser University

mthorogo@sfu.ca

Kıvanç Tatar

Simon Fraser University

ktatar@sfu.ca

Philippe Pasquier

Simon Fraser University

pasquier@sfu.ca

ABSTRACT

Sound designers routinely mix source soundscape recordings. Previous studies have shown that people agree with each other on the perceived valence and arousal for soundscape recordings. This study investigates whether we can compute the perceived emotion of the mixed-soundscape recordings based on the perceived emotion of source soundscape recordings. We discovered quantifiable trends in the effect of mixing on the perceived emotion of soundscape recordings. Regression analysis based on the trajectory observation resulted in coefficients with high R^2 values. We found that the change of loudness of a source soundscape recording had an influence on its weight on the perceived emotion of mixed-soundscape recordings. Our visual analysis of the center of mass data plots found the specific patterns of the perceived emotion of the source soundscape recordings that belong to different soundscape categories and the perceived emotion of the mix. We also found that when the difference in valence/arousal between two source soundscape recordings is larger than a given threshold, it is highly likely that the valence/arousal of the mix is in between the valence/arousal of two source soundscape recordings.

1. INTRODUCTION

Audio-based creative practices, such as sound design and soundscape composition, often use recordings to create musical works. A soundscape recording (or field recording) is “a recording of sounds at a given locale at a given time, obtained with one or more fixed or moving microphones” [1]. Often, sound designers select source soundscape recordings and carefully mix them together, which has a profound influence on meaning, significance, and perceived emotion. Together, the mixed-soundscape recordings create a rich, cohesive experience.

Previous studies demonstrate that people have a high level of agreement on the perceived emotion of source soundscapes recording [2]. It is also possible to build machine-learning models to predict the perceived emotion of soundscape recordings [3]. However, to our knowledge, no study has been presented regarding the effect of mixing on the perceived emotion of soundscape recordings.

In this study, we focus on the effect of mixing on the

perceived emotion of soundscape recordings. We used Emo-Soundscapes, a dataset for soundscape emotion recognition that contains a group of annotated source soundscape recordings and annotated mixed-soundscape recordings [4]. The source soundscape recordings are selected following Schafer’s taxonomy so as to cover the diversity of soundscapes as much as possible [5]. The perceived emotion is represented as the ranking of a two-dimensional vector of valence and arousal [6]. As identified by Thorogood and Pasquier [3], valence represents the pleasantness of a stimulus, which is used to report the perceived pleasantness of a soundscape recording. Arousal indicates the level of eventfulness.

Next, we convert the annotators’ rankings to ratings and used regression models to determine the effect of mixing on the perceived emotion of soundscape recordings. Moreover, we analyzed the center of mass data plots to find the relationships between the perceived emotion of the mixed-soundscape recordings and perceived emotion of source soundscape recordings that are selected within Schafer’s category. Last, we analyzed the likelihood of the perceived emotion of mixed-soundscape recordings lying between the perceived emotions of the two source soundscape recordings that are used for the mix.

2. RELATED WORKS

2.1 Taxonomy of Emotion and Affect Models

Emotional responses are subjective with people having possibly a different response to the same stimulus. According to previous studies [7], two types of emotions are involved when listening to soundscapes:

- *Perceived emotion*: emotions that are communicated and expressed by the source.
- *Induced emotion*: emotional reactions that the source provokes in an audience; it is what the audience feels from the source.

The perceived emotion is the emotion a source expresses. For example, the perceived emotion of happy songs is always “happy”. However, the induced emotion is more subjective. The same happy music may not necessarily induce happiness because of the internal interpretations and experiences of a listener. In this study, we focus on the perceived emotion of soundscapes.

2.2 Soundscape Emotion Studies

There are few studies that investigate modeling the perceived emotion of soundscape recordings. Thorogood and Pasquier [3] propose the Impress system, which uses a linear model predict the perceived pleasantness and eventfulness for soundscape recordings. Building on that research, Fan et al. [8] describe a corpus of audio files extracted from the Sound Ideas sound effects library and the World Soundscape Project library using an automatic segmentation algorithm [9]. A protocol maps audio features and expert user responses to soundscape recordings with stepwise linear regression models. Analysis of the protocol revealed a good fit of features for predicting eventfulness ($R^2: 0.816$) and pleasantness ($R^2: 0.567$). To further the design and evaluation of soundscape emotion research, Fan et al. designed a crowdsourcing listening experiment to collect ground truth annotations of 1,213 audio excerpts [4]. The authors used a ranking-based annotation method instead of rating-based methods [10]. The authors introduced baseline models and defined protocols to assess such models performance. The results of using support vector regressions are human competitive (eventfulness, $R^2: 0.855$; pleasantness, $R^2: 0.629$).

Lundén et al. [11] investigated another method of predicting the outcome of the soundscape assessment based on acoustic features. The authors extracted 120 excerpts (30 seconds) from 77 audio recordings (15 min) and asked 33 participants to move an icon into a 2D space to assess the pleasantness and eventfulness of soundscapes. The authors used the bag-of-frames approach [12] to represent the audio features. Then, they used a Gaussian mixture model to cluster the aggregate of features and used the resulting dissimilarity matrix to train two separate support vector regression models to predict soundscapes' pleasantness and eventfulness. The result indicates that the Mel-frequency cepstral coefficients (MFCCs) provide the strongest prediction for both eventfulness ($R^2: 0.83$) and pleasantness ($R^2: 0.74$).

2.3 Soundscape Taxonomy

Based on Fan et al. [2], we selected sound excerpts following Murray Schafer's soundscape taxonomy [5]. Schafer's referential taxonomy is widely used for the classification of soundscapes. Table 1 shows Schafer's taxonomy.

Categories	Examples
Natural sounds	Bird, thunder, rain, wind
Human sounds	Laugh, whisper, shouts
Sounds and society	Party, concert, store
Mechanical sounds	Engine, factory
Quiet and silence	Quiet part, silent forest
Sounds as indicators	Clock, church bells

Table 1. Murray Schafer's Taxonomy [2, 5].

3. DATASET

We use the Emo-Soundscapes dataset curated by Fan et al. [4]. Emo-soundscapes is a soundscape recording database for soundscape emotion recognition composed of 1213 soundscape excerpts downloaded from Freesound.org. The dataset also contains rankings of the perceived emotion of 1213 6-seconds long soundscape recordings in the 2D valence-arousal space. Fan et al. conducted a crowdsourcing study where 1182 trusted annotators from 74 countries did pairwise comparisons of all soundscape experts regarding perceived valence and perceived arousal. Each pair has been annotated by three annotators. Based on the pairwise comparisons, the database is sorted along the valence and arousal axis.

There are two sets in the Emo-Soundscapes dataset. The first set has 600 excerpts that are selected following Schafer's taxonomy [5] with 100 excerpts per category. The second set contains 613 excerpts that are mixed from the first set. We used the second subset in this study. As described in Tables 2, each mix consists of two or three audio excerpts selected within and between Schafer's soundscape categories. In this paper, we only focus on the mixed-soundscape recordings that are composed of two source excerpts. Before the mixing, each source excerpt is digitally attenuated by either -6 dB or -12 dB. We applied these attenuation levels to examine the influence of loudness changes of sources on the perceived emotion of the mix.

To examine mixing of different types of sounds, we mix excerpts as pairs both from within and between Schafer's categories. Table 2. shows the treatment given to these mixed pairs.

Categories	Excerpt Attenuation (A, B)	Number of Excerpts
Within Soundscape Categories	-6 dB	60
	-12 dB	60
	-6 dB	60
Between Soundscape Categories	-6 dB	75
	-12 dB	75
	-6 dB	75

Table 2. Mixed Audio Excerpts (Two Excerpts) [4].

4. RESULTS AND ANALYSIS

4.1 Regression Analysis

We performed regression analysis on the data. We aim not to maximize absolute performance, but rather to study the relationship between the perceived emotion of two source soundscape recordings and the perceived emotion of the mix, and analyze the influence of the loudness of one source soundscape recording on its weight for the perceived emotion of the mixed-soundscape recording.

We convert the rankings to ratings by mapping the range of ranking values, 1 to 1213, to a range of rating values, 1.0 to -1.0 , so that the highest ranked excerpt has the highest rating. This procedure has two assumptions. First, the distances between two successive rankings are

equal. Second, the valence and arousal are in the range of $[-1.0, 1.0]$. We assumed that two dimensions are independent, and we hypothesized a linear relationship where soundscape recording A and soundscape recording B combine to yield a mixed-soundscape recording. The relationship is as follows:

$$\text{MixedSound}_{\text{Affect}} = \alpha \cdot A_{\text{Affect}} + \beta \cdot B_{\text{Affect}} \quad (1)$$

The subscript “*Affect*” is the dimension (arousal/valence) of emotion. A_{Affect} is the value of affect of soundscape recording A . B_{Affect} is the value of affect of soundscape recording B . $\text{MixedSound}_{\text{Affect}}$ is the value of affect of the mix. α and β are the weights optimized by the regression model, respectively.

We use the coefficient of determination (R^2) to evaluate the performance of our models. R^2 describes the ratio of the variance of the model’s predictions to the total variance. The closer R^2 is to 1, the better the performance of the model. We obtained the R^2 based on 10-fold cross-validation. The results are summarized below.

Dimension	Excerpt Attenuation (A, B)	α	β	R^2
Arousal	−6dB, −6dB	0.597	0.518	0.751
	−12dB, −6dB	0.429	0.612	0.716
	−6dB, −12dB	0.668	0.276	0.724
Valence	−6dB, −6dB	0.535	0.359	0.647
	−12dB, −6dB	0.291	0.590	0.444
	−6dB, −12dB	0.576	0.288	0.526

Table 3. Regression results of predicting the perceived emotion of the mix (Within Schafer’s categories).

Dimension	Excerpt Attenuation (A, B)	α	β	R^2
Arousal	−6dB, −6dB	0.667	0.476	0.660
	−12dB, −6dB	0.483	0.577	0.659
	−6dB, −12dB	0.786	0.353	0.739
Valence	−6dB, −6dB	0.527	0.401	0.216
	−12dB, −6dB	0.496	0.521	0.418
	−6dB, −12dB	0.771	0.271	0.514

Table 4. Regression results of predicting the perceived emotion of the mix (Between Schafer’s categories)

From Tables 3 and 4, we can find correlations between change of loudness and change of weight. When the loudness of soundscape recording A goes from -6 dB to -12 dB, its weight goes down as well. Meanwhile, even though the loudness of the soundscape recording B stays at -6 dB, the weight of soundscape recording B goes up. The same pattern can be found when the loudness of the soundscape recording B goes down and the loudness of the soundscape recording A stay still. The correlation

indicates that the loudness of a soundscape recording has a strong influence on its weight for the perceived emotion of the mixed-soundscape recording.

Comparing the performance of regression models for valence and arousal, we find that the prediction of arousal is more accurately modeled than the valence, confirming the findings in Fan et al. [2].

In general, the results of predicting the valence and arousal of mixed sound within soundscape categories are better than the results of predicting the mixed sound between soundscape categories. Specifically, when both excerpts are attenuated by -6 dB, the results of predicting the valence and arousal of mixed sound within soundscape categories are significantly better than the results of predicting the valence and arousal of mixed sound between soundscape categories. We believe this is because the texture of the mixed-soundscape recordings within the same categories is more homogenous. When mixing them together, it introduces less contrast so that the perceived emotion is more predictable.

4.2 Center of Mass Plots of Mixing Two Source Soundscape Recordings (Within Categories)

In Figures 1–4, we illustrate the center of mass of two source soundscape recordings for visual analysis of the all the data points of mixed sound within soundscape categories. Each figure has 10 mixed-soundscape recordings (combinations of 5 attenuated source recordings) of one soundscape category showing one attenuation condition; adding up to 180 mixed-soundscape recordings.

On these center of mass charts, each green dot represents a source soundscape recording; red stars represent mixed-soundscape recordings. Finally, the influence of the mixing is shown as the trajectory through data points from a source soundscape recording (green circle) through a mixed-soundscape recording (red star) to another source soundscape recording (green circle).

Figure 1 shows the center of mass plots of mixed sound within the categories of “natural sounds” and “quiet and silence.” From Figure 1, we see that when source soundscape recordings have a low level of arousal and a high level of valence, it is the same for the mixed-soundscape recording.

Figure 3 shows the center of mass plots of mixed sound within the categories of “mechanical sounds.” It indicates that when the source soundscape recordings have a high level of arousal and a low level of valence, it is highly likely the case for the mix.

Figure 2 shows the center of mass plots of mixed sound within the categories of “human sounds” and “sounds and society.” In comparison to “natural sounds,” “quiet and silence,” and “mechanical sounds,” the distribution of soundscape recordings on the two-dimensional emotion space is more scattered and the valence/arousal values are more diverse.

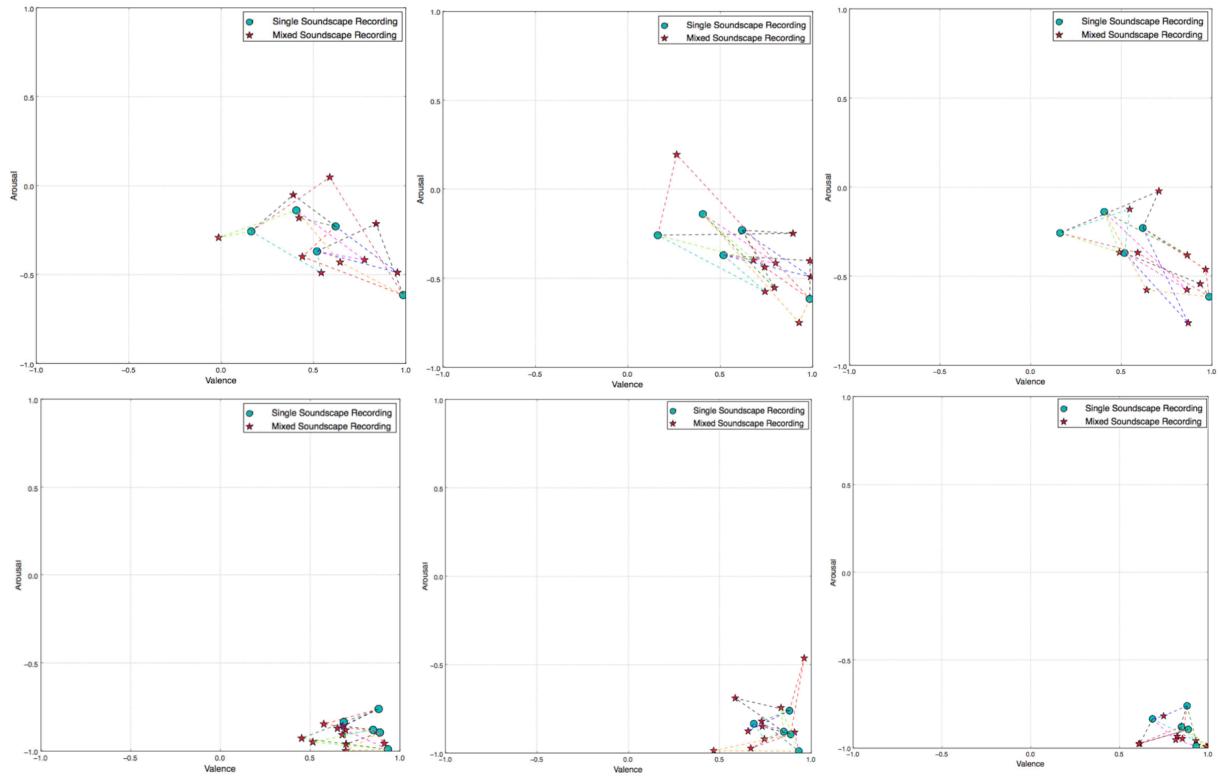


Figure 1. Center of mass of mixed “natural sounds” (Top) and mixed “quiet and silence” (Bottom). The left column shows the attenuation of -6 dB and -6 dB. The middle shows the attenuation of -6 dB and -12 dB. The right column shows the attenuation of -12 dB and -6 dB. (Arousal is the Y-axis, Valence is the X-axis)

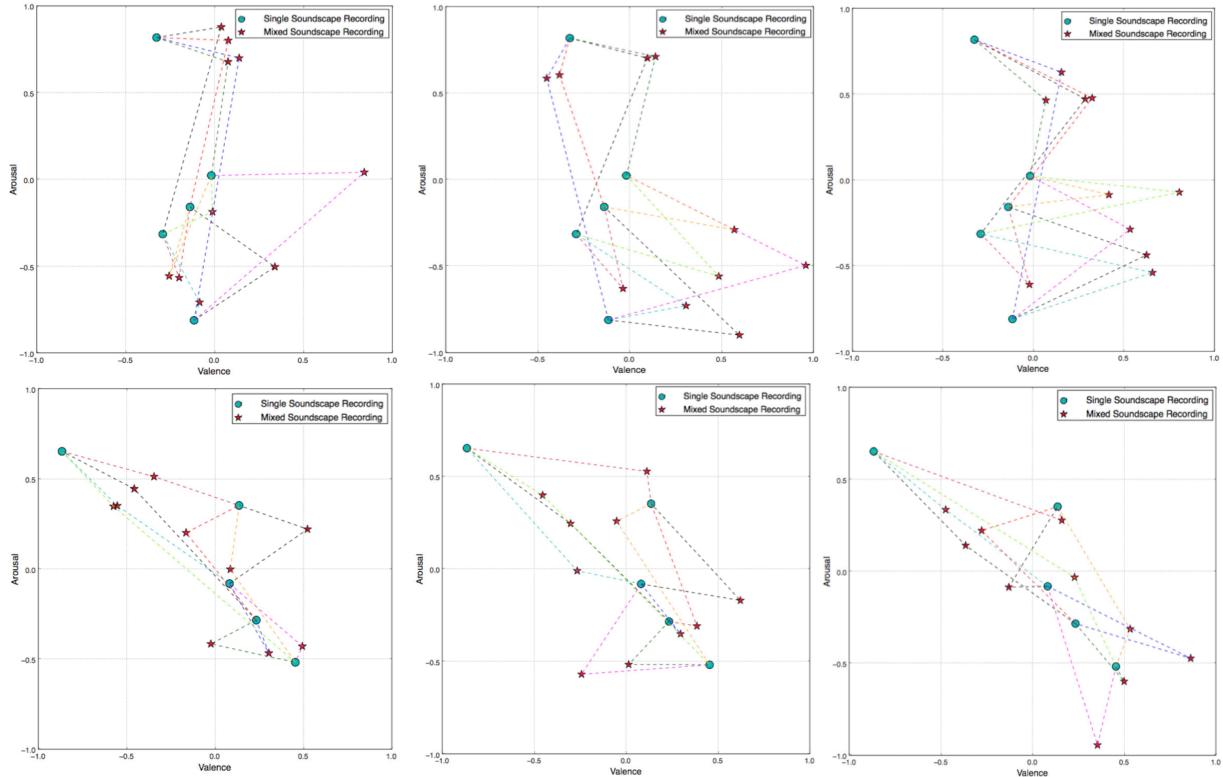


Figure 2. Center of mass of mixed “human sounds” (Top) and mixed “sounds and society” (Bottom). The left column shows the attenuation of -6 dB and -6 dB. The middle shows the attenuation of -6 dB and -12 dB. The right column shows the attenuation of -12 dB and -6 dB. (Arousal is the Y-axis, Valence is the X-axis)

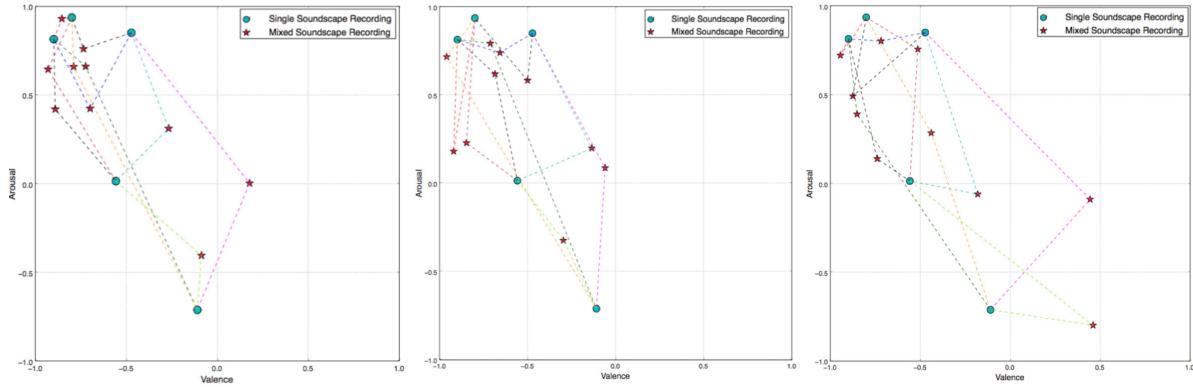


Figure 3. Center of mass of mixed “mechanical sounds”. The left chart shows the attenuation of -6 dB and -6 dB. The middle chart shows the attenuation of -6 dB and -12 dB. The right chart shows the attenuation of -12 dB and -6 dB.

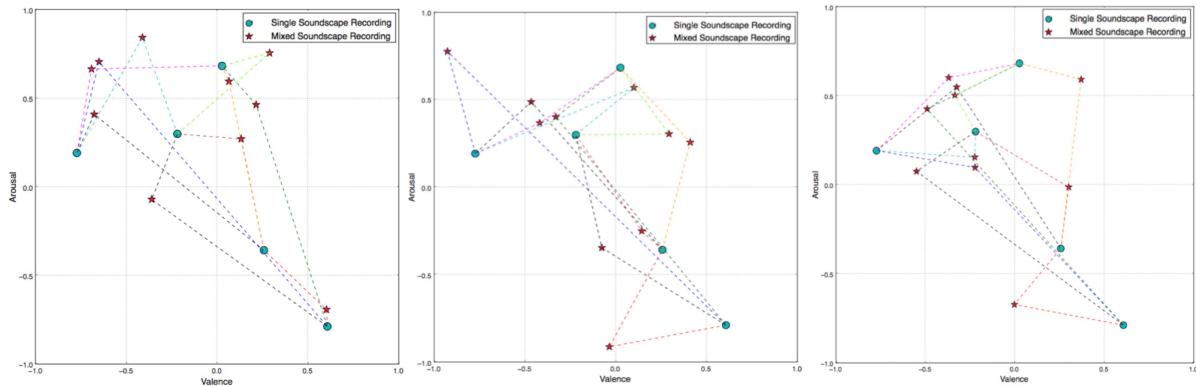


Figure 4. Center of mass of mixed “sounds as indicators”. The left chart shows the attenuation of -6 dB and -6 dB. The middle chart shows the attenuation of -6 dB and -12 dB. The right chart shows the attenuation of -12 dB and -6 dB.

From Figure 2, we can find that the soundscape recordings that have high arousal or low valence (located in the third quadrant) usually have a bigger impact on the valence and arousal of the mixed-soundscape recording, especially when the difference between two source soundscape recordings regarding valence/arousal is large. Mixed soundscapes within the “human sounds” category (-6 dB, -6 dB), for instance, are an example of the hypothesized effect of one source soundscape recording influencing the emotion of the mix. One possible explanation is that the emotion of the high-arousal and low-valence soundscape recordings drew listeners’ attention from the mixed-soundscape recordings.

Moreover, for “human sounds” and “sounds and society,” note that the center of mass for the mixed-soundscape recording is situated on a path between two source soundscape recordings. With only a few exceptions, the mixed-soundscape recording’s valence/arousal ratings lie on a smooth trajectory from one source’s rating to another source’s rating.

Regarding “sounds as indicators,” the relationship between mixed-soundscape recordings’ ratings and source soundscape recordings’ ratings is more complex. Figure 4 shows the center of mass plots for “sounds as indicators.” The fact that “sounds as indicators” is difficult to model confirms the finding in the previous study [2]. “Indicators

serve as clues that something more fundamental or complicated is happening than what is measured by them” [13]. They carry strong semantic information, which is important for perceived emotions.

When we converted rankings to ratings, we also found the following. When the difference between arousal of two soundscape recordings’ ratings is larger than a given threshold (0.5), it is highly likely that the rating of arousal of the mixed-soundscape recording lies in between the rating of arousal of two source soundscape recordings. This is also true for valence. This corresponds to the finding that the mixed-soundscape recording occurs on a trajectory from one soundscape recording to another one. Tables 5 and 6 show the probability of occurrence of the above statement for mixed sound within soundscape categories and mixed sound between soundscape categories.

We also tested the probability of occurrence of the above statement when we removed the soundscape recordings that belong to “sounds as indicators.” Table 5 shows the results, which indicate that the probability increases when we removed “sounds as indicators.” This means the patterns in “sounds as indicators” are more complex. A similar explanation for this is that the semantic information increases the complexity of modeling this category.

Dimension	Excerpt Attenuation (A, B)	Probability	Probability (Not include “sounds as indicators”)
Arousal	−6dB, −6dB	80.00%	89.47%
	−12dB, −6dB	84.00%	89.47%
	−6dB, −12dB	84.00%	89.47%
Valence	−6dB, −6dB	92.86%	100.00%
	−12dB, −6dB	71.43%	87.50%
	−6dB, −12dB	78.57%	87.50%

Table 5. The probability that the rating of the perceived emotion of the mix lies between the ratings of the perceived emotion of sources that are selected within Schaffer’s categories.

Dimension	Excerpt Attenuation (A, B)	Probability
Arousal	−6dB, −6dB	82.67%
	−12dB, −6dB	88.24%
	−6dB, −12dB	84.31%
Valence	−6dB, −6dB	75.56%
	−12dB, −6dB	68.89%
	−6dB, −12dB	73.33%

Table 6. The probability that the rating of the perceived emotion of the mix lies between the ratings of the perceived emotion of sources that are selected between Schafer’s categories.

5. CONCLUSION

We analyzed the relationship between the perceived emotion of mixed-soundscape recordings and source soundscape recordings that are used for mixing. Our analysis shows that there is a correlation between the loudness of a source soundscape recording and its weight that contributes to the perceived emotion of the mix. From the center of mass charts, we found the consistency of perceived emotion of source soundscape recordings and the perceived emotion of mixed-soundscape recording under certain circumstances. Moreover, when the difference of perceived emotion is larger than a given threshold, we found that there is a high likelihood that the perceived emotion of mixed-soundscape recordings lies between the perceived emotions of the two source soundscape recordings that are used for the mix.

The aim of this research is to move toward a formal definition of complex sound design mixing decisions. In doing so, we plan to investigate computational tools that provide suggestions and automate different sound design tasks. One application of this work is in the development of emotion aware digital audio workstations in the production of game sound, film sound, and virtual reality audio environments. We imagine a further integration of such technology in autonomous sound design systems embedded in game engines responding to players’ cues to evoke truly personalized contextual experiences.

6. REFERENCES

- [1] M. Thorogood, J. Fan, and P. Pasquier. “Soundscape Audio Signal Classification and Segmentation Using Listeners Perception of Background and Foreground Sound,” in *Journal of the Audio Engineering Society*, 2016, vol. 64, no. 7/8, pp. 484-492.
- [2] J. Fan, M. Thorogood, and P. Pasquier, “Automatic Soundscape Affect Recognition Using A Dimensional Approach,” in *Journal of the Audio Engineering Society*, 2016, vol. 64, no. 9, pp. 646-653.
- [3] M. Thorogood and P. Pasquier, “Impress: A Machine Learning Approach to Soundscape Affect Classification for a Music Performance Environment,” in *Proc. Int. Conf. on New Interfaces for Musical Expression (NIME2013)*, 2013, pp. 256–260.
- [4] J. Fan, M. Thorogood, and P. Pasquier, “Emo-Soundscapes: A Dataset for Soundscape Emotion Recognition,” in *Proc. Int. Conf. on Affective Computing and Intelligent Interaction (ACII2017)*, 2017.
- [5] R. M. Schafer, *The Soundscape: Our Sonic Environment and the Tuning of the World*, Rochester, VT: Destiny Books, 1993.
- [6] J. A. Russell, A. Weiss and G. A. Mendelsohn, “Affect Grid: A Single-Item Scale of Pleasure and Arousal,” in *J. Personality and Soc. Psych.*, 1989, vol. 57, no. 3, pp. 493–502.
- [7] K. Kallinen, N. Ravaja, “Emotion Perceived and Emotion Felt: Same and Different,” *Musicae Scientiae*, 2006, vol. 5, no. 1, pp. 123-147.
- [8] J. Fan, M. Thorogood, and P. Pasquier, “Automatic Recognition of Eventfulness and Pleasantness of Soundscape,” in *Proc. Audio Mostly*, 2015.
- [9] M. Thorogood, J. Fan and, P. Pasquier, “BF-Classifier: Background/Foreground Classification and Segmentation of Soundscape Recordings,” in *Audio Mostly*, 2015.
- [10] G. N. Yannakakis and H. P. Martínez, “Ratings are Overrated!” *Frontiers on Human-Media Interaction*, 2015.
- [11] P. Lundén, O. Axelsson, M. Hurtig, “On Urban Soundscape Mapping: A Computer can Predict the Outcome of Soundscape Assessments,” in *International Congress and Exposition on Noise Control Engineering: Towards a Quieter Future*, 2016, pp. 4725-4732.
- [12] J. J. Aucouturier and B. Defreville, “Sounds Like a Park: A Computational Technique to Recognize Soundscapes Holistically, Without Source Identification,” in *Proc. Int. Congress on Acoustics*, Madrid, 2007.
- [13] G. H. Orians, M. Dethier, C. Hirshman, A. Kohn, D. Patten, and T. Young, “Sound Indicators: A Review for the Puget Sound Partnership,” *Washington Academy of Sciences*, 2012.

Appendix F

Attending to Breath: Exploring How the Cues in a Virtual Environment Guide the Attention to Breath and Shape the Quality of Experience to Support Mindfulness

MIRJANA PRPA
KIVANÇ TATAR
JULES FRANÇOISE
BERNHARD E. RIECKE
THECLA SCHIPHORST
PHILIPPE PASQUIER

PROCEEDINGS OF THE 2018 DESIGNING INTERACTIVE SYSTEMS CONFERENCE PP.
71-84, ACM PRESS, 2018
[HTTPS://DOI.ORG/10.1145/3196709.3196765](https://doi.org/10.1145/3196709.3196765)

Attending to Breath: Exploring How the Cues in a Virtual Environment Guide the Attention to Breath and Shape the Quality of Experience to Support Mindfulness

Mirjana Prpa

School of Interactive
Arts+Technology, SFU
Surrey, Canada
mprpa@sfsu.ca

Kıvanç Tatar

School of Interactive
Arts+Technology, SFU
Surrey, Canada
ktatar@sfsu.ca

Jules Françoise

LIMSI, CNRS, Univ.
Paris-Sud, Univ. Paris-Saclay,
Orsay, France
jules.francoise@limsi.fr

Bernhard Riecke

School of Interactive
Arts+Technology, SFU
Surrey, Canada
ber1@sfsu.ca

Thecla Schiphorst

School of Interactive
Arts+Technology, SFU
Surrey, Canada
thecl@sfsu.ca

Philippe Pasquier

School of Interactive
Arts+Technology, SFU
Surrey, Canada
pasquier@sfsu.ca

ABSTRACT

Busy daily lives and ongoing distractions often make people feel disconnected from their bodies and experiences. Guided attention to self can alleviate this disconnect as in focused-attention meditation, in which breathing often constitutes the primary object on which to focus attention. In this context, sustained breath awareness plays a crucial role in the emergence of the meditation experience. We designed an immersive virtual environment (iVE) with a generative soundtrack that supports sustained attention on breathing by employing the users' breathing in interaction. Both sounds and visuals are directly mapped to the user's breathing patterns, thus bringing the awareness researched. We conducted micro-phenomenology interviews to unfold the process in which breath awareness can be induced and sustained in this environment. The findings revealed the mechanisms by which audio and visual cues in VR can elicit and foster breath-awareness, and unfolded the nuances of this process through subjective experiences of the study participants. Finally, the results emphasize the important role that a sense of agency and control have in shaping the overall quality of the experience. This can in turn inform the design specifications of future mindfulness-based designs focused on breath awareness.

ACM Classification Keywords

H.5.1. Information Interfaces and Presentation (e.g. HCI): Multimedia Information Systems – Artificial, augmented, and virtual realities

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

DIS'18, June 9–13, 2018, Hong Kong

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM. ISBN 978-1-4503-5198-0/18/06...\$15.00

DOI: <https://doi.org/10.1145/3196709.3196765>

Author Keywords

Mindfulness; Breath Awareness; Virtual Reality; Immersion; Well-Being; Embodiment.

INTRODUCTION

An increasing amount of our daily interactions in the world are mediated by digital technologies. With the rise of ubiquitous computing, an increasing number of our actions are measured, stored and quantified. Smart-phones and wearable technologies track our location, social interactions, and physical activity. The rise of these technologies for the 'quantified self' [77] often focuses on increasing our productivity by tightly monitoring our actions in the world. However, the more we focus on these outward uses of technology, the less we experience an embodied self [70]. Yet, there are opportunities beyond quantification. By designing technologies "as experiences" [40] and by minimizing distractions and bringing focus on the self, we can move toward improving the quality of our interactions and quality of life [5].

In this context, a growing thread of HCI research investigates how embodied approaches to interaction design can support an increased attention to the self [60, 22]. From this body of work, we have focused on designs built upon the concept of *mindfulness*, defined as "the awareness that emerges through paying attention on purpose, in the present moment, and non-judgmentally to the unfolding of experience moment by moment" [26]. Our work builds upon *mindfulness-based design* evolving around the practice of *focused-attention meditation* (FAM) [33]. FAM is a practice of sustaining focused attention on one, "primary", object of attention (internal, such as breathing, or external, such as a candle light). Focused attention on breathing is a widely accepted practice that aims at cultivating a sense of being present with a demonstrated impact on well-being. Engaging in breathing exercises influence cognition, memory, and emotional processing [24, 79, 51], and decreases anxiety and stress [24], even in a young population [30].

A large number of mindfulness-based designs are built upon nuances of FAM with a focus on breathing. Mindfulness-based design can be used to guide a user's attention to breath, for example through Virtual Environments (VEs) that employ breathing cues [64, 52]. Despite a significant number of mindfulness-based designs that guide the attention to breath, our understanding of how breath awareness is elicited and how the design of VEs can sustain the user's attention on breath are still limited. This is what motivates the research effort presented here.

In this paper, we investigate how a continuous interaction between breathing patterns and a virtual environment can induce and support the breath awareness of users. Our inquiry is motivated by the preliminary findings in [54] that showed the potential of the VE and interaction design to elicit and sustain the user's awareness of their breathing. Hence, our goal in this study is to better understand (1) how the users of the *Pulse Breath Water* system become aware of their breathing in a VE without verbal guidance during the interaction, (2) how the cues in the VE can support sustained attention on breathing, and (3) gain a deeper understanding of the actual experience of becoming aware and sustained awareness of one's breath. To answer these research questions we (1) built the *Pulse Breath Water* system, and (2) propose an empirical qualitative study employing micro-phenomenology interviews [50].

BACKGROUND

Mindfulness-Based Design

The third wave in HCI [4] brought the embodied approach to interaction [11] to the forefront by acknowledging the importance of the embodied experience in sense-making of the experience. A number of design approaches position the body at the center of a user's experience. For instance, *Somaesthetics* designs cultivate an aesthetic appreciation of bodily experience [62, 59]. *Somaesthetic Appreciation design* gives rise to the experience through guided attention inwards, to the self; excluding external distractions and making space for somatic awareness to arise [23]. Yet another concept, *Somatic Connoisseurship*, demonstrates how "body-based awareness skills" can be utilized in experience design and designing for the experience of the self, by building upon somatic phenomenology and first-person practices for self-awareness [61].

The designs we focus on are built upon mindfulness principles and encourage a shift of attention inwards, towards bodily sensations, in order to support well-being and self-regulation processes [3]. Mindfulness has been defined as a state of being non-judgmental and present in the moment-to-moment unfolding of the experience [26]. Mindfulness-based designs aim to bring one to the state of mindfulness through various approaches. For example, Zhu et al. [80] recognized four approaches to designing for mindfulness: *Digitalized mindfulness* that is a digital equivalent to guided meditation; *Personalized mindfulness* that addresses personal preferences regarding mindful meditation; *Quantified mindfulness* that is built upon applications that offer real-time sensing and feedback; and *Presence-in and Presence-with* approach that goes beyond tools and offers a way of mindful being rather than doing.

The most common mindfulness-based designs take the form of a mobile application [8] that tackle mindfulness from different angles: mobile applications that guide a user to distance itself from troubling thoughts [78, 47], foster mindful walking through ambient sound generated from walking and breathing patterns [7], applications that serve as mindfulness meditation timers [48], applications for didactic guided meditation sessions [69], or applications that integrate mindfulness principles within a broader framework such as *acceptance and commitment therapy* [1]. Beside mobile and computer applications, mindfulness-based designs have taken on the form of audio installations - *Sonic Cradle* [73, 74], virtual environments [41] - *The Meditation Chamber* [64], *RelaWorld* [32], *Sanctuarium* [12], *PsychicVR* [2] and more recently, mixed reality environments - *Inner Garden* [58].

Focused-Attention Meditation

Mindfulness practices encompass various approaches to bringing the user's awareness to the present moment. A growing number of mindfulness-based designs revolve around the practice of focused-attention meditation (FAM) [37, 38]. In FAM a practitioner shifts their attention from external stressors to internal sensations such as breathing [55], and is considered the most widely accessible practice among novice meditators [38]. Practicing sustained focused attention on breath has been shown to reduce stress and improve well-being [31]. Most importantly, FAM fosters interoceptive awareness, an ability to receive and attend to the signals originating in our bodies [14, 21], which is shown to not only improve attention task performance but as well contribute to emotion regulation [10, 44].

Designs for Breath Awareness

How can we support the meditation community with technology [9] is an open-end question that is explored within the HCI community from many different angles. We focus on designs that "facilitate mindful moments" of being present and self-aware by directing the user's attention to their breathing [55]. We review systems using multi-sensory feedback to support mindful reflection, by "bringing unconscious aspects of experience to conscious awareness" [63]: p.50. Many designs for breath awareness have been proposed in HCI [66]. *Sonic Cradle* [73] was designed for cultivating mindfulness through a soundscape in which sounds are triggered by breathing. *BrightBeat* [18] was designed for cultivating calmness and focus through utilizing screen brightness with breathing patterns and audio feedback to guide a user towards intended breathing. *SomaMat and Breathing Light* demonstrated a different approach to mindfulness through breathing by emphasizing breath-related somaesthetic qualities through light and heat feedback [67].

Specifically, we focus on Virtual Reality (VR) as an "embodied technology" that can support the user's attention to bodily sensations using the sense of immersion and presence [56]. By employing audio-visual cues to breath as another representation cue, VR can provide a sensory-augmentation dimension that supports the user in focusing attention on breath. Because immersion in VEs minimizes external distractions, VR is a promising medium to support mindfulness practice and elicit breath awareness [65, 55].

Different approaches to designing for breath awareness and FAM in virtual reality have been proposed. In *Solar* [52], breathing and electroencephalography (EEG) data are mapped to the elements in the virtual environment that provides audio and visual cues and is presented on a desktop screen. *Guided Meditation VR* [42] and *JunoVR* [75] immerse the user in a VEs that resemble nature and uses audio instructions to guide the user's attention to breathing. Similarly, abstracted natural environments are used as a design element in: *Lumen* [43] that employs guided meditation and gaze interaction for navigation through an enchanted forest; *Life Tree* in which the growth and liveness of the tree are controlled by the user's breathing rhythm [49], and in *Deep* which depicts an underwater fantasy world which the user navigates by deepening their breathing and adjusting to a slow pace [71]. Other applications employ continuous – and potentially ambiguous – feedback to guide the users' attention. The *Meditation Chamber* [64], provides biofeedback in a VE using multimodal data – breathing, electrodermal activity (EDA), and blood volume pressure – along with guided instructions, and presented on HMD. Similarly, in *Strata*, breathing, heart rate, EDA, and brain activity determine the visuals and the audio of the virtual environment depicting five different worlds unified into a single experience [13].

However, while some of these applications are built upon the gamification paradigm [25], and often include guided meditation as an invite to the experience, our interest is in the experiences that can guide user's attention to breathing through the cues in the environment. Therefore, our approach does not involve any guided meditation instructions, nor have we intended for our VE design to be a game. Our focus is on VR applications presented on immersive HMD and without verbal guidance into the practice. Hence, we designed *Pulse Breath Water* as a generative environment in which the events are determined by breathing patterns and the system's decision (by AI agent), where the user's breathing is employed as one and only modality for the cues in the iVE.

Despite a significant number of mindfulness-based VEs that support breath awareness, there is still a lack of a deeper understanding of how specifically breath awareness is elicited through different cues and interactions in VR. In this work, we aim to get a detailed understanding of user experiences of a particular design using micro-phenomenology interviews. Our design employs breathing data in a continuous audio and visual feedback in the virtual environment. The complexity of the visual elements is kept minimal to minimize dispersion of attention. The immersiveness of the medium contributes to decreased external distractions and an ability to elicit body sensations and direct attention towards one's breath, supported by generative audio that responds in real time to the changes in breathing patterns.

Micro-Phenomenology: A Methodology for the Study of Experience

To understand the subjective experiences of the participants, we chose *micro-phenomenology*, an inquiry method developed by Petitmengin [50, 45] upon Vermersch's *Explication interview* technique [72]. The objective of micro-phenomenology is to obtain explicit descriptions of *singular experiences* as

they unfold in a larger, chronological structure of the experience, bypassing generalized typical post-hoc descriptions.

For the descriptions of the experience to emerge in the interview, the interviewer and the interviewee use a precise communication-protocol as a tool for mediating the first-person point of view of the interviewee through the second-person position of the interviewer. The interviewee accesses the experience retrospectively following the guidance of the interviewer who takes a role of a mediator. The interviewer's role is to guide and stabilize the interviewee's attention, guide them through the process of evoking the experience, and then direct their attention to a particular dimension of the experience. Once the interviewee's attention is stabilized and they are in an *evocative state* – a state of re-living the experience – the interviewer guides the interviewee in deepening the description to the required level of precision.

Micro-phenomenology and *Explication interview* have been explored in HCI context in the past. Light [34] discussed the explication interview in the context of gathering experiences of using websites [34] and receiving mobile phone calls [35, 36]. Hogan [20] used the method to gather the users' experiences of data representation. Françoise et al. [15] utilized the method to evaluate the system they built for kinesthetic awareness while the same method was used by Candau et al. [6] to explore how interaction informed by somatic practices and embodied cognition contributes to cultivating kinesthetic awareness. To our knowledge, while the interest in micro-phenomenology to understand user experiences is growing, no published research has used the methodology in VR.

A VIRTUAL ENVIRONMENT DESIGNED TO SUPPORT BREATH AWARENESS

Pulse Breath Water system was initially created as an artwork [53] that evolved and doubled into a research instrument over time. We undertake a research-through-design [16] approach, and design iterations are informed by the exploratory studies we conduct.

Interaction Scenario

The main design premise is that the events in the virtual environment are determined by the breathing patterns. Deep slow breathing triggers slower, more sustained movement on the up/down-axis, allowing users to observe, reflect, and position in a particular part of the environment. Sustained breaths allow for staying in a place, while fast, strong breaths cause erratic movement. Metaphoric mapping [39] of movement allows for interaction that is easy to understand: on the participant's inhale, the position of the participant rises in the environment, and on the exhale they sink, just like when submerged in water. In our VE, the respiration data guides the audio generation in the affective space. The eventfulness of the audio is mapped to the appearance of the waves in the ocean. More eventful breathing generates more eventful audio that then generates a more disturbed ocean surface and increased waves. The element that depicts the passage of time is the sky that changes color from light gray to pitch black within a span of 6 minutes.

System design

Our design brings together three components: (1) A respiration sensor and respiration analysis module that drives (2) an immersive virtual environment (displayed on Oculus Rift SDK2), and (3) a generative sound environment. The novelty of the system, compared to other VR systems presented above, is in the generative audio and the AI agent that determines the events in the environment based on eventfulness of breathing that, to our best knowledge, have not been utilized so far in iVE.

The overall system outline is represented in Figure 1. One respiration sensor (Thought Technology) [76] attached to the user's abdominal area streams respiration data to M+M middleware [46] to MAX MSP¹ patch. The audio is generated by an autonomous agent that selects samples from the audio corpus according to the frequency of the user's breathing. All audio samples are tagged with different eventfulness properties using a state of the art music emotion recognition algorithm [68]. The overall eventfulness values of the audio environment is sent to the 3D game engine Unity 3D² along with respiration data via OSC messages. This data generates visual changes in the VE presented to the user via HMD. The user listens to the audio environment with circumaural noise-canceling headphones. A second respiration sensor was placed around the chest, and was used for data collection and as a reference, however chest data was not employed in the design.

Musical Agent: listening and responding through sound

Musical agents are artificial agents that automatize musical tasks. The hybrid musical agent generates the audio environment using the patterns of the user's breathing: slower subtle breathing triggers the agent to play less eventful sounds, and vice versa, faster deeper breaths result in the agent playing sounds that will be perceived as more eventful. The audio corpus contains piano recordings of musical chords that do not create musical tension and resolution. These recordings are further processed so that the origin of the recordings are not clear to the listeners. The recordings are all labeled with vectors with two dimensions: average pleasantness and average eventfulness based on Music Emotion Recognition algorithm [68]. Using Music Emotion Recognition, the agent maps the breathing to the eventfulness of audio samples to create an interactive sonic environment.

We apply signal processing to the respiration data using a wavelet transform. The wavelet transform outputs 24 frequency bands and these bands are mapped to the audio corpus' eventfulness range. The pleasantness range of the audio samples are substantially smaller than the range of eventfulness. The agent applies a random walk on the pleasantness dimension to create variations on the sample selection. The agent uses the generated eventfulness and pleasantness values to choose samples from the audio corpus. The selected sample is played by one of four playback engines of the agent.

In addition to piano sounds, a wave-table synthesizer [57] is used to generate a heartbeat-like sound controlled by the speed of user's breathing: the pulsation sound in the audio

environment slows down as the user's breathing slows down. The interaction between the user and the audio environment is further enhanced by introducing a low-pass filter. As the user breathes in and out, the timbre of the audio environment oscillates between a muddy, low-frequency prominent audio environment and a full-spectrum audio environment. This also enhances the submersion feeling associated with being under the virtual water surface in the virtual environment (see section below *Virtual Environment: ART and Ambiguity of relationship as an invitation to the experience*).

Lastly, the agent applies Music Emotion Recognition algorithm to estimate the eventfulness and pleasantness of the generated sonic environment. The estimation algorithm is the online version of the estimation algorithm that is used to label the audio in the agent's memory. The output of online affect estimation is a vector with two dimensions: eventfulness and pleasantness. These values are further used to control the parameters of the virtual reality environment.

Virtual Environment: ART and Ambiguity of relationship as an invitation to the experience

Immersive environments can positively affect the user's attention, which was explained by Kaplan in *Attention Restoration Theory (ART)*[29]. ART focuses on the correlation between type of stimuli and restorative potential of nature environments [28]. The environments with stimuli that modestly capture attention are preferred over the stimuli that elicits mental fatigue and cognitive overload, and the design of our system relies on this principle. We used Unity 3D to generate the environment and 3D elements of a body of water - an ocean. The aesthetics of the scene is intentionally minimal, displaying the ocean and the sky in a range of gray-scale shades. The use of color is minimal. Design of elements in the VE is informed by the concept of *beholder's share* [27] in which the user's previous experiences guide the process of meaning-making of ambiguous stimuli. The ambiguity of visual stimulus encourages an interpretative relationship between a user and the environment [17]. In particular, *ambiguity of relationship* [17] between the user and the visual stimulus evokes users to project their values and experiences in reflection and meaning-making. For example, the ambiguous environment can be perceived as frightening and anxiety-inducing by one person or calm that elicits relaxation and peaceful feelings in other, building upon each person's previous real life experiences with real-life environments. Finally, continuous breathing patterns allow users to control the environment and curate their own experience.

Respiration data from an abdominal sensor controls the user's position in the environment. After initial testing, we decided to include movement on a vertical axis because the movement on a horizontal axis (along the ocean surface) was inducing motion-sickness. The mapping was informed by metaphoric mapping [39] built upon cognitive schema of "more is up, less is down (for example)". The more the participant breathes in, the higher in the VE they will go. We limited the height to which participants can move to prevent falling down from a high distance as that was reported as anxiety inducing by pilot participants. The eventfulness level is mapped to waves on the ocean surface: more eventful audio will cause more excited waves, and vice versa.

¹<https://cycling74.com>

²Unity 3D: <https://unity3d.com/>

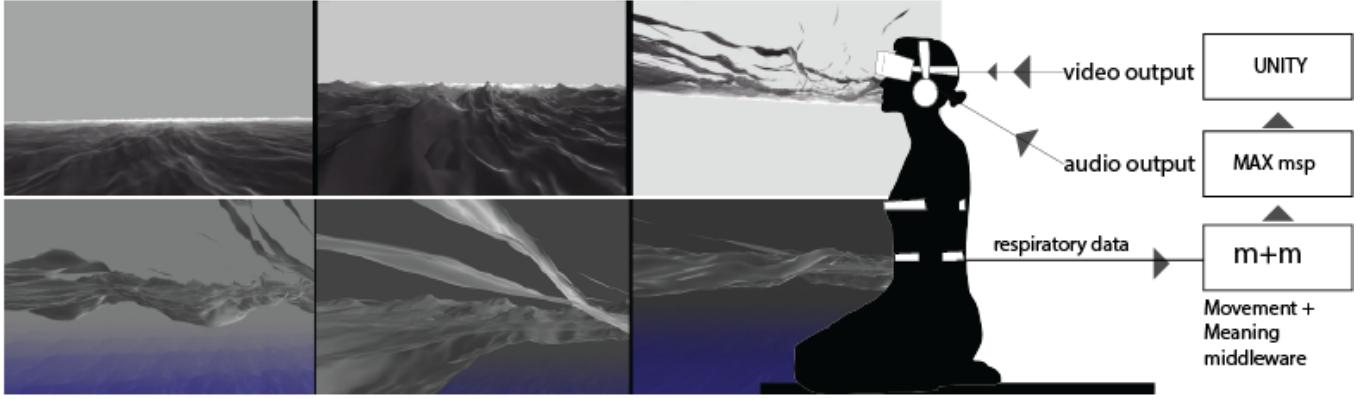


Figure 1. The system design

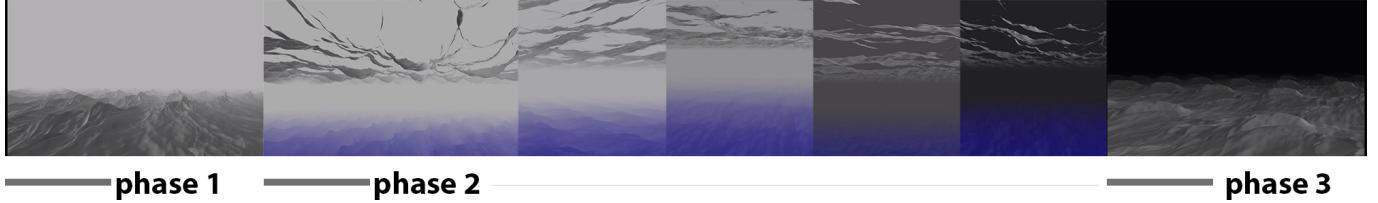


Figure 2. Duration: phase 1 (1 min) starts off with light gray sky and a stationary user's position above the ocean surface level; Phase 2 (4 min) is when the breath to movement mapping in the VE is activated and the user is moving along the vertical axis according to their breathing; the sky colour progresses from gray to black; Phase 3 (1 min)- the user is stationary again, and the only difference between this and phase 1 is in the colour of the sky

USER STUDY: UNDERSTANDING PERSONAL EXPERIENCES THROUGH MICRO-PHENOMENOLOGY

We conducted an exploratory study investigating how audio and visual cues mapped to breathing in the VE give rise and support to sustained breath awareness. The study aims to understand the experience of becoming aware of breath, and the nuances of the system that either support or distract a user in attending to self and becoming breath-aware. The participants were introduced to the VE following a four-part protocol described in more detail in Section *Protocol* (below). The insights are induced using micro-phenomenology interviews [50] after each completed session.

Artwork as a research instrument The initial design of our system was built upon the idea of a real-time generated piece, without narrative, end or beginning. To allow for gradual transition to and out of the environment that would enable us to inquire about the process of eliciting breath awareness, we iterated the system previously created as an artwork *Pulse Breath Water* [53] and added three phases. In the first and the last phase (phases 1 and 3) (see Figure 2), the user has a stationary position above the ocean and is presented with the cues of their agency through an audio environment. In the second phase, the user's agency is reflected not just through the audio (as in phase 1, and 3) but also their breathing controls the vertical position in the environment, moving the user above the ocean's surface with breath in and below the surface with breath out. The third phase is the same as the first phase, and the user is back to the starting position above the water while exposed to the cues of their breathing through the audio.

Designing 3 phases allowed for a variety of cues of the user's agency in the VE that could have influenced different mechanisms for eliciting breath awareness that we aimed to reveal in the interviews. The main points of interest were the switches between the phases in which the user's breath started impacting the environment (first reflected through audio cues in phase 1, then increasing the agency in phase 2 by adding movement cues, and then decreasing agency in phase 3 by presenting the audio cues only).

The length of the phases was determined in a pilot study conducted with 6 participants. The criteria was to find the balance between the time that seemed to keep the user's interest and flow state, and the point when the familiarity and lack of events in the environment becomes dull. Taking these pilot sessions into account, we reduced the length of the phases without agency as the users reported them as dull. Finally, the first phase is 60 seconds long, the second phase is 240 seconds long, and finally, the third phase is 60 seconds adding to the VE's duration to 6 minutes total (see Figure 2).

Participants

We recruited 11 participants (7 female) through the university mailing list and social media channels. Participants' ages ranged from 24 to 44 (mean: 27.1, SD: 5.87). Two participants had never tried VR before, eight participants had been exposed to it less than ten times overall, and one is an expert VR user. Regarding mindfulness practice, three participants meditate regularly, one has never meditated, and the rest meditate irregularly. All participants reported good health condition and normal vision.



Figure 3. The participant interacting with the VE, wearing breathing sensors, headphones, and Oculus Rift while seated on the cushion

Apparatus and Data collection

The participants were seated comfortably on a large bean bag pillow (see Figure 3), one at the time, in a dark room, at the computer station. The VE was presented on Oculus SDK2 at the rate of 90 FPS. The audio component of the VE was played on noise-canceling headphones. Participants wore respiration sensors (Thought technology [76]) positioned on the abdomen and chest and that data was captured. Also, interviews were audio and video recorded.

Protocol

Upon arrival, the participants read a description of the study. After agreeing to participate by signing the consent form, participants were equipped with two breathing sensors: abdominal and chest, and Oculus Rift, and the noise-canceling headphones. Prior to the sessions, we informed all participants that: “the virtual environment is reacting to your breathing” and we invited the participants to explore the experience without disclosing mapping details. The session was divided into four parts (see Figure 4):

1. *Session 1: Exploration*: The participants interacted with the virtual environment using their breathing for a duration of 6 minutes.
2. *Did the participant understand control and mapping details?*: After Session 1, we started interviews with all participants. Very early in these interviews we were able to determine whether participants made sense of the interaction and if they understood the correlation between the breath and the movement and sonic events in the environment (i.e., if they understood the mapping).
3. *Interview*: If they did, we would proceed to a full-length micro-phenomenology interview (this was the case with 3 participants³). If they did not make this connection (8 participants), they were instead invited to proceed with Session 2 followed by the full-length interview.

³we will elaborate on this further in the Discussion section

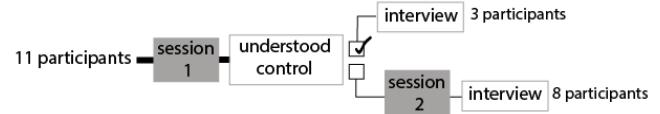


Figure 4. the illustration of the protocol

4. *Session 2: disclosed mapping details*: The 8 participants who did not make a sense of the control they had over the cues in the environment via their breathing, were explained the mapping details, that is how their breathing influences the virtual visual and sonic environment. Finally, they were asked to explore the environment again for another 6 minutes going through the 3 phases already described.
5. *Interview*: After the second session we conducted micro-phenomenology interviews, each 30 minutes in length.

Analysis

All interviews were transcribed and coded using the nVivo software for qualitative analysis. All interview data was structured in chronological order, and the descriptions of singular experiences were identified. The coding was done in two stages: first, interviews were coded line-by-line. Our coding system in this stage included the codes: control, attention, breath, body sensations, imagery, sound, feeling, inner voice, and were informed by six pilot interviews that we conducted prior to the study. In the second stage of coding, we identified common themes from the codes identified in the first stage of coding. Two researchers (the authors number 1 and 5), both trained in conducting micro-phenomenology interview, discussed and agreed on the themes.

FINDINGS

Unfolding the process of eliciting breath awareness

Attention shifts in becoming aware of breath

The complexity of the environment and the novelty of the experience can potentially be overwhelming to the participants. However, the three phases of the experience offered the possibility for a gradual transition of the focus from the environment to the bodily sensations related to the movement, and finally to the breath. Once aware of their control of the elements in the environment, a majority of the participants would direct attention first to the waves, the vertical movement patterns, and then to changes in the sonic environment, thus gradually discovering the nuances of the control they had over the environment, before starting to focus their awareness on their breathing. However, participants with meditation experience tended to immediately focus on the breath before shifting their attention to the environment, and finally to their bodily sensations:

At the very beginning I was focusing on my breath, just trying to listen and understand what was happening. And then the second part because I got the connection instead of listening to my own breath I changed my breath playing with the images and then I went to first state of listening to my body but I wasn't anxious anymore about motion sickness and the storms in the ocean and... I was focusing on really listening. (P17)

Focused attention

Once familiarized with the environment, participants directed their attention to preferred parts in the experience by practicing controls they experientially learned during the session:

Initially I was repeating it, long breath in out in out but I didn't hold my breath even inside. After some time when it went on for some cycles then I started different things different rhythm of my breath as if I am controlling it. I was just trying to control that up and down... after that I realized [that] inside I'm feeling good, then I started staying inside. (P14)

My focus is just I guess on the tranquility and lack of sharp noises and you know... how much air do I need to stay... like when should I take my next breath, and you know if I take my next breath is it going to be really deep breath. (P9)

Finally when transitioning to phase 3 and losing the movement on a vertical axis, their attention shifted back to their body and “letting go”:

OK, now I'm controlling my breath, and then with the white peaceful situation I attuned to my body and that was it, I wasn't controlling. I let that go... (P17)

Cues in the VE for eliciting breath awareness

One of the most common difficulties for beginners in mindfulness breath FAM is attending to breathing without being distracted. In our design, this process was supported by the visual representation of breathing as a movement in the environment. As a matter of fact, participants were expected to first direct their attention outwards to the environment in order to be able to redirect it to the self. Their vertical position being controlled by how full their chest was simply allowed them to literally visualize their breathing, thus expliciting it.

I was more conscious of it [of breath] because of what I was seeing at the same time. [How did that make you conscious of breath?] I wasn't really concentrating only on my breath but also on what I saw... so I knew that I was breathing in but also by seeing where I was... or... at what point I was by seeing where I was. (P10)

Figure 5 depicts the moment in which participant 17 became aware of the relationship between breath and the changes in the environment. The plot shows a change in their breathing pattern after the realization of their agency and control of the environment. The participant described this change as:

After a couple [breaths] I was like “no that's me, if I'm holding my breath I see down, every time I'm exhaling I'm down there, and every time I'm inhaling I'm up there” and then I tried to hold my breath both in the inhalation and exhalation and I was like “Hey, yay, I got it”. (P17)

Regarding the audio cues, the majority of participants made a connection between the sound and the breath through the visual cues of their position in the environment:

When I'm staying level there [on the surface]... the sound gets a little bit violent I guess. It was just like, there was some sort of something like chords going on



Figure 5. Change in breathing pattern following realization of how the participant's breathing influences the system

with some dissonance pattern maybe bit when I was level there I felt like there something transcendental creature got real angry about what I was doing and like the sound got violent and even although I enjoyed being level there for a while I felt “for this sound to change I have to move” and I went back to breathe in breathe out pattern again. (P15)

Finally, some participants related the perceived motion cues to perceived bodily sensations of the breath. After the initial “a-ha” moment of breath awareness, the motion cues caused participants to be more aware of the range, rhythm, depth, and other subtleties of their breathing and the other bodily sensations that accompany inhalation and exhalation.

I paid attention to correlation and inhaling... feeling my body move, and also like the airflow through my nose, and my mouth... I feel like when my body became still I inhaled completely, and there is moment of no movement then that's when the motion stopped. (P11)

Breathing patterns and perceived sense of agency

The comparison of breathing data of Session 1 (participants unaware of agency) and Session 2 (participants aware of the agency and interacting with the environment) revealed different breathing patterns and the qualities of the experience associated with the interaction. This is depicted in Figures 6 and 7. The upper images of both Figure 6 and Figure 7 reveal that when these two participants were not aware of their agency in the environment (were not aware of the changes their breathing was causing to the environment), the breathing patterns remained somewhat consistent within all three phases. However, the bottom images demonstrate that the changes in the phases are followed by the changes in the breathing patterns, indicating that the perceived agency in the environment in this case, influenced how the user interacted with the environment (leading them to take on more active role) and their experience of it.

The first one I feel like because I didn't know that I had any control over what was happening I think it was lot more chaotic in my mind. And the second one was lot more calming because I felt I had that control. (P10)

An interesting finding is that the sense of agency in some participants defined the perceptive position of the participant. In the first trials, those who were not aware of the direct control they had of the system were more likely inclined to take a passive role in the environment similar to the “observer's perspective” or that the system was pushing them up and down.

In the first one I was thinking that something is pushing me up and down like bouncing... this time I was thinking

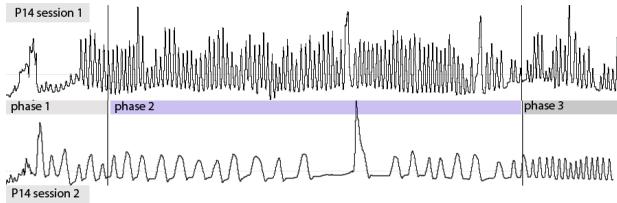


Figure 6. breathing data during Session 1: no perceived sense of agency vs breathing data in the Session 2: perceived agency (bottom image).

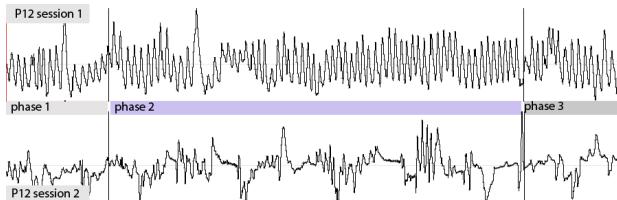


Figure 7. (upper image) breathing data in the Session 1 vs breathing data in the Session 2 (bottom image)

I am controlling this and I can calm this ocean... so I was feeling this was inside my body, it's not the system, it's inside of me. (P12) (see Figure 7)

Creating the experience from “within”

Once the participants became aware of their breath, of the agency that their breath has in the environment and how they can control the experience, they started exploring the environment. After participants got familiar with the elements of the environment and identified the most pleasurable parts they started directing the experience from “within”, through their breathing. Curating the experience through agency and perceived control over the environment helped the users to leverage the initial personal differences in initial reaction to the environment, while gradually familiarizing with the control they had in the environment.

Personal differences in experiencing ambiguous VE

Previous, real-life experiences influenced participants’ experience of the environment; participants who fear water made negative associations with our environment and vice versa, participants who relate positive experiences to being in the water found the environment more pleasurable.

I didn't like the feeling of going underneath... probably not because of this experience but probably related to other experiences with water and because it looks a lot like water... [how does this make you feel] it's like a little bit of anxiety... or like fear in a way but not really fear because I know I can breathe. (P10)

Following the evocation of previous experiences, the immersiveness of the medium triggered strong bodily sensations related to the overall quality of the experience.

I did experience that feeling of euphoria as I was coming up and and traveling upwards I did get that sensation of actually moving up and when I looked down I just really felt compelled this was overwhelming feeling for me to

wanna go deeper... and I just wanted so hard to get rid of all of my breath so I could go deeper. (P13)

It was kind of like swimming and holding a breath underneath the water... and then like I could feel that physical aspect to it. (P7)

Perception of control alters the experience

A few participants initially experienced VE as anxiogenic or fear-inducing. However, they alleviated these initial negative associations once they realized that they were in control.

I was like “oh I’m in control actually” and that’s very comfortable, because that anxiety went away straight away... it’s very comfortable that anxiety was out of it... I can stay under the water as long as I want. That’s totally fine – I told myself... and then I also see the patterns below and then I felt this urge to explore it. (P15)

Curating the experience from a conscious control of a breath

In the second phase, once the participants were aware of their breath and control that they had over the environment, they would purposefully manipulate their breathing patterns to either stay in a preferred location in the environment or avoid potentially unpleasant stimuli.

I kind of found myself holding my breath when I breathe out because I liked the blue better the description of the VE... so I found myself kind of breathing out lot more than what I was inhaling that's what I was observing when I was really focused on my breathing. (P16)

Some of the participants were able to identify moments when their breathing changed as a response to the environment, which induced a reaction. The reactions to irregular breath varied from focusing on breath closely to trying to change the position in the environment to a preferred one.

In that first phase I'd say I think I was just attuning to the environment. [how?] I listened to my breath and I realized because I'm not doing well, I suffer of sea sickness, so I was "I don't know if this is going work" and I realize that my breath was going faster than what would normally happen, but then I just tried to listen to my breath and that was pretty much what I did. (P17)

Associating irregularity of breath with a particular part of the environment that helped participants know how to regulate breath when it becomes irregular:

Then I was asking myself: ok now I'm breathing quite heavily so... uhm... can I also maybe just try to get back to the white plane again? (P8)

Subtle and unaware of influence of the environment on breathing patterns

In Session 1, only three participants were immediately aware of their agency. The data plots of the remaining eight participants indicate that there is an influence of the system on their breathing patterns. This influence is particularly clear in the moments of phase changes, revealing two trends:

1. *Transient perturbation of breath by changes in the mapping*
Figure 8 depicts the respiration sensor data for participant 16 in

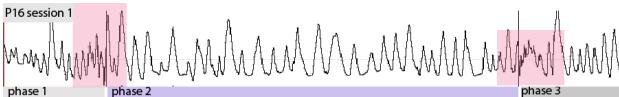


Figure 8. Respiration sensor data for participant 16 in the first session. Although P16 was not aware that their breath controlled the visual feedback, we observe shifts in breathing patterns following the changes between the three phases.

the first session. Although P16 was not aware that their breath controlled the feedback, we observe that transitions from phase 1 to phase 2 and from phase 2 to phase 3 were followed by changes in breathing patterns. Even though participants were not aware of the fact that they influenced the system, the data suggests that changes in the mapping of breath to the visual feedback induced perturbations of the participants' breathing patterns.

2. Sustained Influence of the mapping on the breathing patterns

Finally, the comparison of breathing patterns across phases showed that participants' breathing patterns changed on a longer time scale throughout the session. For example, the breathing period of P16 significantly changed between phase 1 and phase 2, as highlighted in Figure 8. In phase 2 participants tended to breathe at a slower rate, extending inhalation and exhalation over longer periods of time compared to phase 1.

DISCUSSION

In this section, we discuss the findings in the context of designing for breath awareness. This discussion is informed by participants' accounts about the process of becoming aware of their breath, and by our design decisions.

The role of sense of agency and control in overall experience

Understanding the control and engaging in the interaction impacts the quality of the experience and thus influences the experience-driven designs [12]. In our study protocol, we allowed participants to explore our system in Session 1 without giving away the details about how the system reacts to their breathing. Eight participants did not make an explicit connection between their breathing and changes in the environment in Session 1. We speculate that this happened due to an overall novelty with virtual reality applications, the lack of knowledge regarding the respiratory sensor, and that once immersed in the environment the participants were engaged in sense making of the scene rather than raising focus and awareness to the interaction. That could possibly explain why some of the participants reported that "something" [P12] was pushing them up and down, without realizing that that "something" was their breathing. Feeling of being dependent on the system while immersed in the VE caused unpleasant feelings that were alleviated once the participants became aware of their agency on the system. We find this to be a challenge to be addressed in future work. How we can design systems that make users aware of their agency in a subtle, less instructed way is yet to be addressed.

Interestingly, three participants did make an immediate connection between their breath and the environment. After the interview, when asked whether they meditate, and how often, those three participants reported that they regularly meditate. While we are careful not to draw strong conclusions from this observation, it has been shown that meditators have more accurate visual attention compared to non-meditators [19, 37] and this might explain why those three participants made an immediate connection.

In conclusion, while the system did not change, perceived sense of agency and control over the system determined the qualities of the experience: if the sense of agency was not perceived, the experience was more likely to be unpleasant. Understanding how the system could be controlled was crucial in a further unfolding of the experience. Participants exercised control especially in the parts of the environment that were more likely to induce fear and anxiety. At signs of discomfort caused either by the movement or audio-visual feedback, participants consciously changed their breathing to influence changes in the environment towards creating a more pleasant experience.

Complexity of the VE mediated through gradual introduction of the elements through phases

During the design process we were faced with the decision of how much feedback is optimal for eliciting breath awareness while preventing possible distractions. Through an iterative design process we decided to gradually increase the threshold for the stimulus, introducing sound as the first cue, and then movement as the second cue to breathing. This allowed participants to familiarize themselves with the environment before introducing more stimulating, and more obvious cues such as the movement along the vertical axis.

The majority of the participants described the up-down movement as the most dominating cue to one's breathing in our environment. One explanation might be that of the immersiveness of VR as a medium that can elicit bodily sensations related to locomotion. Moving on the vertical axis up and down was described as being on the roller coaster or a swing. In the second phase, the majority of participants' reports about breathing were related to the movement rather than sound. Sound as a cue was discussed within the first and third phase, when the movement cues were absent. The absence of the movement allowed the participants to listen and "attune" to their breath:

[in the end the third phase] I was just watching and I heard a sound but my attention and my focus was inside and just listening what was happening... in the second [the second phase] I was still doing that but because I knew I could control what I was watching instead of attuning to my breath I was using my breath to change what I was observing. (P17)

Despite our attempts to understand an optimal amount of cues to guide the user's attention through the pilot study from which we drew insights for the final design presented here, some of the participants' accounts imply that a few things distracted them from fully attending to breathing. The main distractions

are related to the interaction design: the velocity of the movement on the vertical axis was reported as too fast. Since the mapping of the movement was not elaborate in this iteration of the system, we see the potential of fine-tuning the mapping of the movement and better controlling for the velocity. We speculate that fine-tuning of the movement parameters will provide more opportunities for variety in how users interact with the system and how breath awareness can be raised in more subtle movements in the environment, and this will be explored in our future work.

Attuning to the environment

While we expected to see the changes in breathing patterns within different phases in Session 2 when the participants were aware of their agency and how they control the changes in the environment, unexpected findings emerged from Session 1. Even though the participants were not aware of the differences between the three phases, nor were they aware of how they influence changes in the environment, breathing data revealed that breathing patterns changed in participants (9 out of 11 participants) following shifts between phase 1 to phase 2 to phase 3. This suggests that our virtual environment influenced the breathing patterns through implicit mechanisms, and we are curious to explore this further.

Meditation tool

As a part of inquiry after the micro-phenomenology interview, we asked the participants: “If we tell you that this VE is a tool, what would you use this tool for?”. The majority of the participants responded that they would use it to: overcome a fear of water, reflect, isolate themselves from distractions, or meditate. One of the participants even compared the breath-driven control to breathing exercises often done as an intro to mindfulness meditation practice:

it felt... it's kind of when you are meditating and you do breath work before. And so I felt kind of the same in terms I wasn't using breathing technique but to say "ok now I'm controlling my breath" and then with the white peaceful situation I attuned to my body and that was it, I wasn't controlling I let that go. (P17)

Presence of visual cues helped the participants focus on their breath which is commonly reported as an obstacle in novice meditators:

You can pay more attention to the visual or what's in front of you in environment to make that difference in your body instead of having to pay attention to your breathing to affect your breathing, you pay attention to the environment to affect your breathing. (P10)

Traditional mindfulness meditation requires mediators to maintain some kind of *meta-awareness* so they can notice once their mind wanders off and they lose focus on their breathing and gently guide their attention back to their intended point of focus. This need to maintain a meta-awareness constitutes one of the main meditation challenges and obstacles especially for novice meditators. Similar to the playful interaction paradigm in the auditory environment of *Sonic Cradle* [73, 74], the audio-visual virtual environment in the current study was designed to subtly and unobtrusively help users to re-focus their

attention to their breathing once they lost it – as if the system would simulate the meta-awareness that the users might not have developed yet. Instead of requiring users to maintain meta-awareness, our system was designed to help users reorient their focus in a more playful manner, which we hope will eventually help them to more easily re-direct focus in their everyday life.

Commentary on the methodology: Can we trust the participant's descriptions?

In this study, the interviewer has been trained in the method, but none of our participants have been subject to a micro-phenomenology interview prior to participating in our study. While no method to our knowledge can provide us with conclusive evidence that participant's actual experience matches their description of it, we found some signs of such matchings. When we compared participants' interviews with their breathing data, it showed that the descriptions explained the breathing data such as changes in breathing rhythm patterns when participants reported that they changed their breathing, or holding the breath to stay in a particular part of the environment, or sudden deep inhales/exhales showed in plots. Despite the difficulties that some of the participants experienced with an evocation of the experience, we are confident in the validity of the descriptions because their descriptions can explain the breathing data and the breathing data corroborates their descriptions. However, we are yet to explore what is the level of detail in the descriptions that we can obtain from participants and how fine-grained descriptions correlate with the breathing data.

CONCLUSION

Our motivation was to gain a deeper understanding of how the experience of becoming aware of breath in VE unfolds. Previous mindfulness-based designs that employ breath demonstrated the important role that breathing and breath awareness hold in mindfulness practices. However, the understanding of the process of how people become aware of breath through embodied interaction was largely missing and the goal of this study was to take us a step closer to that knowledge. Our contribution is threefold: First, we presented the design of a system that employs breathing as embodied interaction in VE for eliciting breath awareness; Second, we conducted a micro-phenomenology inquiry and unveiled the mechanisms of becoming aware of breath; Third, we presented the findings that revealed the process of becoming aware of breathing in VE and demonstrated the importance of a sense of agency, understanding of control and possible subtle impact of the environment on breathing patterns without users being aware of it. In addition, some of the initial unpleasant experiences and personal differences in the quality of how participants experienced VE can be alleviated if users are enabled to curate their experiences. Finally, we contribute to the research on micro-phenomenology and HCI by revealing that participants' descriptions of their experience can describe their breathing data. This indicates the potential of micro-phenomenology and motivates us to continue to explore this methodology paired with physiological data in future work.

ACKNOWLEDGMENTS

The authors would like to thank all the volunteers for participating in the study, Ash Tanasiychuk for tireless proofreading, and the reviewers for their valuable feedback. We would like to acknowledge that this research would not be otherwise possible without the grant support from The Social Sciences and Humanities Research Council of Canada (SSHRC), and equipment support from Natural Sciences and Engineering Research Council (NSERC).

REFERENCES

1. Aino Ahtinen, Elina Mattila, Pasi Väkkynen, Kirsikka Kaipainen, Toni Vanhala, Miikka Ermes, Essi Sairanen, Tero Myllymäki, and Raimo Lappalainen. 2013. Mobile mental wellness training for stress management: feasibility and design implications based on a one-month field study. *JMIR mHealth and uHealth* 1, 2 (2013).
2. Judith Amores, Xavier Benavides, and Pattie Maes. 2016. Psychicvr: Increasing mindfulness by using virtual reality and brain computer interfaces. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 2–2.
3. Kirk Warren Brown and Richard M Ryan. 2003. The benefits of being present: mindfulness and its role in psychological well-being. *Journal of personality and social psychology* 84, 4 (2003), 822.
4. Susanne Bækker. 2006. When Second Wave HCI Meets Third Wave Challenges. In *Proceedings of the 4th Nordic Conference on Human-computer Interaction: Changing Roles (NordiCHI '06)*. ACM, New York, NY, USA, 1–8. DOI: <http://dx.doi.org/10.1145/1182475.1182476>
5. Rafael A Calvo and Dorian Peters. 2014. *Positive computing: technology for wellbeing and human potential*. MIT Press.
6. Yves Candau, Jules Françoise, Sarah Fdili Alaoui, and Thecla Schiphorst. 2017. Cultivating kinaesthetic awareness through interaction: Perspectives from somatic practices and embodied cognition. In *Proceedings of the 4th International Conference on Movement Computing*. ACM, 21.
7. Sixian Chen, John Bowers, and Abigail Durrant. 2015. 'Ambient Walk': A Mobile Application for Mindful Walking with Sonification of Biophysical Data. In *Proceedings of the 2015 British HCI Conference (British HCI '15)*. ACM, New York, NY, USA, 315–315. DOI: <http://dx.doi.org/10.1145/2783446.2783630>
8. Luca Chittaro and Andrea Vianello. 2014. Computer-supported Mindfulness: Evaluation of a Mobile Thought Distancing Application on Naïve Meditators. *Int. J. Hum.-Comput. Stud.* 72, 3 (March 2014), 337–348. DOI: <http://dx.doi.org/10.1016/j.ijhcs.2013.11.001>
9. Katie Derthick. 2014. Understanding Meditation and Technology Use. In *CHI '14 Extended Abstracts on Human Factors in Computing Systems (CHI EA '14)*. ACM, New York, NY, USA, 2275–2280. DOI: <http://dx.doi.org/10.1145/2559206.2581368>
10. Anselm Doll, Britta K Hözel, Satja Mulej Bratec, Christine C Boucard, Xiyao Xie, Afra M Wohlschläger, and Christian Sorg. 2016. Mindful attention to breath regulates emotions via increased amygdala–prefrontal cortex connectivity. *NeuroImage* 134 (2016), 305–313.
11. Paul Dourish. 2004. *Where the action is: the foundations of embodied interaction*. MIT press.
12. Laura L Downey. 2015. *Well-being technologies: Meditation using virtual worlds*. Ph.D. Dissertation. Nova Southeastern University.
13. Isabelle Du Plessis. 2017. Strata: a biometric VR experience. In *ACM SIGGRAPH 2017 VR Village*. ACM, 14.
14. Norman Farb, Jennifer Daubenmier, Cynthia J Price, Tim Gard, Catherine Kerr, Barnaby D Dunn, Anne Carolyn Klein, Martin P Paulus, and Wolf E Mehling. 2015. Interoception, contemplative practice, and health. *Frontiers in psychology* 6 (2015), 763.
15. Jules Françoise, Yves Candau, Sarah Fdili Alaoui, and Thecla Schiphorst. 2017. Designing for Kinesthetic Awareness: Revealing User Experiences through Second-Person Inquiry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, 5171–5183.
16. William Gaver. 2012. What Should We Expect from Research Through Design?. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 937–946. DOI: <http://dx.doi.org/10.1145/2207676.2208538>
17. William W. Gaver, Jacob Beaver, and Steve Benford. 2003. Ambiguity As a Resource for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '03)*. ACM, New York, NY, USA, 233–240. DOI: <http://dx.doi.org/10.1145/642611.642653>
18. Asma Ghandeharioun and Rosalind Picard. 2017. BrightBeat: Effortlessly Influencing Breathing for Cultivating Calmness and Focus. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1624–1631.
19. Holley S Hodgins and Kathryn C Adair. 2010. Attentional processes and meditation. *Consciousness and cognition* 19, 4 (2010), 872–878.
20. Trevor Hogan. 2015. Tangible Data, a Phenomenology of Human-Data Relations. In *Proceedings of the Ninth International Conference on Tangible, Embedded, and Embodied Interaction (TEI '14)*. ACM Press, Stanford, CA, USA, 425–428. DOI: <http://dx.doi.org/10.1145/2677199.2691601>
21. Britta K Hözel, Sara W Lazar, Tim Gard, Zev Schuman-Olivier, David R Vago, and Ulrich Ott. 2011. How does mindfulness meditation work? Proposing mechanisms of action from a conceptual and neural perspective. *Perspectives on psychological science* 6, 6 (2011), 537–559.

22. Kristina Höök, Martin P Jonsson, Anna Ståhl, and Johanna Mercurio. 2016. Somaesthetic Appreciation Design. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, San Jose, CA, USA, 3131–3142. DOI: <http://dx.doi.org/10.1145/2858036.2858583>
23. Kristina Höök, Martin P. Jonsson, Anna Ståhl, and Johanna Mercurio. 2016. Somaesthetic Appreciation Design. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 3131–3142. DOI: <http://dx.doi.org/10.1145/2858036.2858583>
24. Ravinder Jerath, Molly W Crawford, Vernon A Barnes, and Kyler Harden. 2015. Self-regulation of breathing as a primary treatment for anxiety. *Applied psychophysiology and biofeedback* 40, 2 (2015), 107–115.
25. Daniel Johnson, Sebastian Deterding, Kerri-Ann Kuhn, Aleksandra Staneva, Stoyan Stoyanov, and Leanne Hides. 2016. Gamification for health and wellbeing: A systematic review of the literature. *Internet Interventions* 6 (2016), 89–106.
26. Jon Kabat-Zinn. 2003. Mindfulness-Based Interventions in Context: Past, Present, and Future. *Clinical Psychology: Science and Practice* 10, 2 (2003), 144–156. DOI: <http://dx.doi.org/10.1093/clipsy.bpg016>
27. Eric Kandel. 2016. *Reductionism in Art and Brain Science: Bridging the Two Cultures*. Columbia University Press.
28. Stephen Kaplan. 1995. The restorative benefits of nature: Toward an integrative framework. *Journal of environmental psychology* 15, 3 (1995), 169–182.
29. Stephen Kaplan. 2001. Meditation, Restoration, and the Management of Mental Fatigue. *Environment and Behavior* 33, 4 (July 2001), 480–506. DOI: <http://dx.doi.org/10.1177/00139160121973106>
30. Kiat Hui Khng. 2016. A better state-of-mind: deep breathing reduces state anxiety and enhances test performance through regulating test cognitions in children. *Cognition and Emotion* (2016), 1–9.
31. Bassam Khoury, Tania Lecomte, Guillaume Fortin, Marjolaine Masse, Phillip Therien, Vanessa Bouchard, Marie-Andrée Chapleau, Karine Paquin, and Stefan G Hofmann. 2013. Mindfulness-based therapy: a comprehensive meta-analysis. *Clinical psychology review* 33, 6 (2013), 763–771.
32. Ilkka Kosunen, Mikko Salminen, Simo Järvelä, Antti Ruonala, Niklas Ravaja, and Giulio Jacucci. 2016. RelaWorld: neuroadaptive and immersive virtual reality meditation system. In *Proceedings of the 21st International Conference on Intelligent User Interfaces*. ACM, 208–217.
33. Tatia MC Lee, Mei-Kei Leung, Wai-Kai Hou, Joey CY Tang, Jing Yin, Kwok-Fai So, Chack-Fan Lee, and Chetwyn CH Chan. 2012. Distinct neural activity associated with focused-attention meditation and loving-kindness meditation. *PLoS One* 7, 8 (2012), e40054.
34. Ann Light. 2006. Adding method to meaning: A technique for exploring peoples' experience with technology. *Behaviour & Information Technology* 25, 2 (2006), 175–187. DOI: <http://dx.doi.org/10.1080/01449290500331172>
35. Ann Light. 2008. Transports of delight? What the experience of receiving (mobile) phone calls can tell us about design. *Personal and Ubiquitous Computing* 12, 5 (2008), 391–400.
36. Ann Light. 2009. Negotiations in space: The impact of receiving phone calls on the move. *The reconstruction of space and time: Mobile communication practices* (2009), 191–213.
37. Dominique P Lippelt, Bernhard Hommel, and Lorenza S Colzato. 2014. Focused attention, open monitoring and loving kindness meditation: effects on attention, conflict monitoring, and creativity—A review. *Frontiers in psychology* 5 (2014).
38. Antoine Lutz, Heleen A Slagter, John D Dunne, and Richard J Davidson. 2008. Attention regulation and monitoring in meditation. *Trends in cognitive sciences* 12, 4 (2008), 163–169.
39. Anna Macaranas, Alissa N Antle, and Bernhard E Riecke. 2015. What is intuitive interaction? balancing users' performance and satisfaction with natural user interfaces. *Interacting with Computers* 27, 3 (2015), 357–370.
40. John McCarthy and Peter Wright. 2004. Technology As Experience. *interactions* 11, 5 (Sept. 2004), 42–43. DOI: <http://dx.doi.org/10.1145/1015530.1015549>
41. A Relaxing Virtual Reality Application / Guided Meditation. 2017. (2017). <http://guidedmeditationvr.com/about-2/>
42. Cubicle Ninjas. Virtual Reality Meditation. 2016. (2016). <http://guidedmeditationvr.com/about-2/>
43. Lumen VR meditation. 2016. (2016). <https://www.viveport.com/apps/c666dd8d-e42b-403c-91d3-58693f268220>
44. W-E Mehling. 2001. The experience of breath as a therapeutic intervention—psychosomatic forms of breath therapy. A descriptive study about the actual situation of breath therapy in Germany, its relation to medicine, and its application in patients with back pain. *Complementary Medicine Research* 8, 6 (2001), 359–367.
45. Claire Petitmengin : Micro-phenomenology. 2017. (2017). <https://www.microphenomenology.com/>
46. Movement and Meaning | Canarie Middleware. 2017. (2017). <http://www.mplusm.ca/>
47. Just Let Go on the App Store. 2011. (2011). <https://itunes.apple.com/us/app/just-let-go/id439683407?mt=8>

48. Lotus Bud Mindfulness Bell on the App Store. 2012. (2012). <https://itunes.apple.com/us/app/lotus-bud-mindfulness-bell/id502329366?mt=8>
49. Rakesh Patibanda, Florian 'Floyd' Mueller, Matevz Leskovsek, and Jonathan Duckworth. 2017. Life Tree: Understanding the Design of Breathing Exercise Games. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*. ACM, 19–31.
50. Claire Petitmengin. 2006. Describing one's subjective experience in the second person: An interview method for the science of consciousness. *Phenomenology and the Cognitive Sciences* 5, 3-4 (2006), 229–269. DOI: <http://dx.doi.org/10.1007/s11097-006-9022-2>
51. Pierre Philippot, Gaëtane Chapelle, and Sylvie Blairy. 2002. Respiratory feedback in the generation of emotion. *Cognition & Emotion* 16, 5 (2002), 605–627.
52. Mirjana Prpa, Karen Cochrane, and Bernhard E. Riecke. 2015. Hacking Alternatives in 21st Century: Designing a Bio-Responsive Virtual Environment for Stress Reduction. In *Pervasive Computing Paradigms for Mental Health*, Silvia Serino, Aleksandar Matic, Dimitris Giakoumis, Guillaume Lopez, and Pietro Cipresso (Eds.). Number 604 in *Communications in Computer and Information Science*. Springer International Publishing, 34–39. http://link.springer.com/chapter/10.1007/978-3-319-32270-4_4 DOI: 10.1007/978-3-319-32270-4_4.
53. Mirjana Prpa, Kivanç Tatar, Philippe Pasquier, and Bernhard Riecke. 2016. Pulse Breath Water. <http://madzie.com/portfolio/pulse-breath-water/>. (2016). [Online; accessed March 2018].
54. Mirjana Prpa, Kivanç Tatar, Bernhard E Riecke, and Philippe Pasquier. 2017. The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality. In *International Conference on Virtual, Augmented and Mixed Reality*. Springer, 153–172.
55. Cassandra L Reese. 2017. *Breath in Motion: Breath Awareness Design Research Study*. Ph.D. Dissertation. Kent State University.
56. Giuseppe Riva, Rosa M Baños, Cristina Botella, Fabrizia Mantovani, and Andrea Gaggioli. 2016. Transforming experience: the potential of augmented reality and virtual reality for enhancing personal and clinical change. *Frontiers in Psychiatry* 7 (2016).
57. Curtis Roads. 1996. *The Computer Music Tutorial*. The MIT Press, Cambridge, Mass.
58. Joan Sol Roo, Renaud Gervais, and Martin Hachet. 2016. Inner garden: An augmented sandbox designed for self-reflection. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction*. ACM, 570–576.
59. Thecla Schiphorst. 2009a. soft(n). In *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI EA '09)*. ACM, Boston, MA, USA, 2427. DOI: <http://dx.doi.org/10.1145/1520340.1520345>
60. Thecla Schiphorst. 2009b. *The Varieties of User Experience: Bridging Embodied Methodologies from Somatics and Performance to Human Computer Interaction*, Ph.D. dissertation. Ph.D. dissertation. Simon Fraser University. http://www.sfu.ca/~tschiph0/PhD/PhD_thesis.html
61. Thecla Schiphorst. 2011. Self-evidence: applying somatic connoisseurship to experience design. In *CHI'11 extended abstracts on human factors in computing systems*. ACM, 145–160.
62. Richard Schusterman. 2012. Somaesthetics definition. In *encyclopedia of Human-Computer Interaction. Interaction Design Foundation*.
63. Phoebe Sengers, Kirsten Boehner, Shay David, and Joseph 'Jofish' Kaye. 2005. Reflective Design. In *Proceedings of the 4th Decennial Conference on Critical Computing: Between Sense and Sensibility (CC '05)*. ACM, New York, NY, USA, 49–58. DOI: <http://dx.doi.org/10.1145/1094562.1094569>
64. C. Shaw, D. Gromala, and A. Fleming Seay. 2007. The meditation chamber: Enacting autonomic senses. *Proc. of ENACTIVE 7* (2007), 405–408. <http://www.sfu.ca/~shaw/papers/Enactive07MedChamber.pdf>
65. Jacek Sliwinski, Mary Katsikitis, and Christian Martyn Jones. 2015. Mindful gaming: how digital games can improve mindfulness. In *Human-Computer Interaction*. Springer, 167–184.
66. Jacek Sliwinski, Mary Katsikitis, and Christian Martyn Jones. 2017. A Review of Interactive Technologies as Support Tools for the Cultivation of Mindfulness. *Mindfulness* (2017), 1–10.
67. Anna Ståhl, Martin Jonsson, Johanna Mercurio, Anna Karlsson, Kristina Höök, and Eva-Carin Banka Johnson. 2016. The Soma Mat and Breathing Light. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 305–308.
68. Kivanç Tatar and Philippe Pasquier. 2017. MASOM: A Musical Agent Architecture based on Self Organizing Maps, Affective Computing, and Variable Markov Models. In *Proceedings of the 5th International Workshop on Musical Metacreation (MUME 2017)*. Atlanta, Georgia, USA.
69. The Mindfulness App: Guided & Silent Meditations to Relax on the App Store. 2012. (2012). <https://itunes.apple.com/us/app/mindfulness-app-guided-silent/id417071430?mt=8>
70. Sherry Turkle. 2011. Alone together. (2011).
71. Marieke Van Rooij, Adam Lobel, Owen Harris, Niki Smit, and Isabela Granic. 2016. DEEP: A biofeedback virtual reality game for children at-risk for anxiety. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. ACM, 1989–1997.

72. Pierre Vermersch. 2009. Describing the Practice of Introspection. *Journal of Consciousness Studies* 16, 10-12 (2009), 10–12.
73. Jay Vidyarthi and Bernhard E. Riecke. 2013. Mediated Meditation: Cultivating Mindfulness with Sonic Cradle. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*. ACM, New York, NY, USA, 2305–2314. DOI: <http://dx.doi.org/10.1145/2468356.2468753>
74. Jay Vidyarthi, Bernhard E Riecke, and Diane Gromala. 2012. Sonic Cradle: designing for an immersive experience of meditation by connecting respiration to music. In *Proceedings of the designing interactive systems conference*. ACM, 408–417.
75. Juno VR. 2017. (2017). <http://junovr.com/blog/2017/2/11/a-breathing-sensor-for-virtual-reality>
76. Thought Technology Ltd. ProComp2 2 Channel Biofeedback & Neurofeedback System w/ BioGraph
77. Infiniti Software Thought Technology Ltd. 2017. (2017). <http://thoughttechnology.com>
78. Gary Wolf. 2010. The data-driven life. *The New York Times* 28 (2010), 2010.
79. Throw your worry away! 2011. (2011). http://cdn.appshopper.com/icons/437/253188_larger.png
79. Christina Zelano, Heidi Jiang, Guangyu Zhou, Nikita Arora, Stephan Schuele, Joshua Rosenow, and Jay A Gottfried. 2016. Nasal respiration entrains human limbic oscillations and modulates cognitive function. *Journal of Neuroscience* 36, 49 (2016), 12448–12467.
80. Bin Zhu, Anders Hedman, and Haibo Li. 2017. Designing Digital Mindfulness: Presence-In and Presence-With Versus Presence-Through. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 2685–2695. DOI: <http://dx.doi.org/10.1145/3025453.3025590>

Appendix G

The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality

MIRJANA PRPA
KIVANÇ TATAR
BERNHARD E. RIECKE
PHILIPPE PASQUIER

AS PUBLISHED IN THE PROCEEDINGS 19TH INTERNATIONAL CONFERENCE ON HUMAN-COMPUTER INTERACTION, 2017

The Pulse Breath Water System: Exploring Breathing as an Embodied Interaction for Enhancing the Affective Potential of Virtual Reality

Mirjana Prpa, Kivanç Tatar, Bernhard E. Riecke, and Philippe Pasquier

School of Interactive Arts and Technology, Simon Fraser University
[mprpa, ktatar, b_r, pasquier]@sfu.ca

Abstract. We introduce *Pulse Breath Water*, an immersive virtual environment (VE) with affect estimation in sound. We employ embodied interaction between a user and the system through the user’s breathing frequencies mapped to the system’s behaviour. In this study we investigate how two different mappings (metaphoric, and “reverse”) of embodied interaction design might enhance the affective properties of the presented system. We build on previous work in embodied cognition, embodied interaction, and affect estimation in sound by examining the impact of affective audiovisuals and two kinds of interaction mapping on the user’s engagement, affective states, and overall experience. The insights gained through questionnaires and semi-structured interviews are discussed in the context of participants’ lived experience and the limitations of the system to be addressed in future work.

Table of Contents

Abstract	1
Introduction	2
1 Background	3
1.1 Embodied interaction and embodied cognition in interactive systems	3
1.2 Unconventional interfaces	4
Conventional vs unconventional interfaces	4
Breath-controlled Interfaces	4
1.3 Affect and Sound Estimation	5
2 Pulse Breath Water, an immersive virtual environment with affect estimation in sound	6
2.1 Interaction Scenario	6

2.2	System Description	7
2.3	<i>PBW</i> as an Art Installation	9
3	Methodology	10
3.1	Study design	10
3.2	Conditions and mappings	10
3.3	Participants	11
3.4	Experimental setup	11
3.5	Procedure	11
3.6	Data collection	12
3.7	Data analysis	12
4	Quantitative findings	12
4.1	Questionnaire findings	12
4.2	Affect Grid and STAI-6	14
5	Interviews	14
5.1	Exploring the unknown: phases	14
5.2	Regardless of mapping, second trial is enjoyed more	14
5.3	Metaphoric mapping feels intuitive, but reverse is more playful ..	15
5.4	Somatic experiencing: awareness of breath and emerging past experiences through the changes in the environment	15
5.5	Loss of control triggers fear?	16
5.6	Imagine this was a tool...	16
6	Discussion	16
6.1	Familiarity first, engagement after	17
6.2	Tension and relaxation, at the same time	17
6.3	The context is the key	17
7	Conclusion	18
8	Acknowledgements	18

Introduction

We react to the environments we inhabit, and these environments can have an immense impact on us. The physical environments are not the only environments we interact with. The progress in VR technology since 2010 demonstrated the potency of virtual reality (VR) to make us feel present in virtual environments (VEs). The sense of presence that we feel in VEs is sometimes so strong that we react to the events in VEs as we would react to them in physical reality [30]. Many authors argue that one of the crucial determinants to make us feel present in a certain environment is the ability of that environment, physical or virtual, to support our actions [38,15,31]. We use our whole bodies and engage them in more or less subtle movements to perform many of our actions, and interact with the environments or engage in talking, walking, or even breathing. The importance of understanding interaction as a whole body activity was recognized by the third wave in human-computer interaction (HCI) research that emphasized an embodied approach to the design of interactive systems. Dourish [9] defines

embodied interaction as an approach to interaction design that emphasizes the integrity of our minds and bodies engaged in actions with environments, in a process through which meaning and understanding are generated. Along with this idea, Slater and Sanchez-Vives [31] argued that even perception is “a whole body action”. A “whole body interaction” is defined as “The integrated capture and processing of human signals from physical, physiological, cognitive and emotional sources to generate feedback to those sources for interaction in a digital environment” [12].

We focus here on the idea that our thinking and learning processes originate in our bodies as much as in our brains (the concept from embodied cognition). In particular, we are focusing on subtle body movements engaged in the process of breathing and mapping the abdominal movement to the changes in our virtual environment (VE). Given the connection between cognition and the body, we ask: if VEs feel “real” and are perceived as real, physical environments, *how can these environments change us through embodied interaction design? And, how can an embodied interaction design that employs the user’s subtle breathing movements facilitate these changes?* These overarching questions motivated the research presented here. We speculate that the ways in which our environments can shape us depends on the environment’s properties. In this paper, we investigate how the type of interaction mapping supported by an affect estimation of previously recorded sounds enhances the affective properties of the VE. For our test-bed we employed *Pulse Breath Water*¹ (PBW), an immersive virtual environment presented on head-mounted display-HMD (Oculus Rift DK2). The interaction between a user and the environment is enabled through the user’s breathing patterns that generate changes in the virtual environment.

PBW has been publicly shown as an art installation; we gained insights from the audience that motivated the research presented here. By undertaking an embodied interaction design research approach and mixed methods, we investigated how different mappings (metaphoric, and “reverse”) between the user’s breathing patterns and the system response (reflected by the changes in audio and visual components) can be used to influence the user’s affective state, user’s engagement and overall user experience.

1 Background

1.1 Embodied interaction and embodied cognition in interactive systems

The shift in HCI paradigms that emphasized embodiment followed the shift in cognitive research. For a long time it was understood that “thinking” happens in our heads only; however, the emerging body of research on embodied cognition argues that body interactions with the environment is the basis for cognitive processes [37]. In other words, to understand the world around us, we use not only our brains, but our bodies too.

¹ <http://ispace.iat.sfu.ca/project/pulse-breath-water/>

In particular, research on embodied interaction draws an understanding of interaction processes from situated bodies, minds, and the environment. Dourish [9] used the term “embodied interaction” to explain the embodied nature of an interaction in physical and social contexts. We interact with others and physical objects in the environment, and through these “embodied actions” we make meanings [9]. Grounded in phenomenology, Robertson [26] focuses on Merleau-Ponty’s teachings, and speculates that in the centre of embodied interaction is a living body, our tool to experience the surrounding world. Similarly, the design of interactive systems should follow the interaction principles we establish with the world around us [9].

Similarly to Dourish, Kirsh [18] focuses on the lens of embodied cognition to emphasize the potential for designing interactive systems. Their premise is that when we manipulate objects we shape our concepts and beliefs through the actions that employ those objects in the environment. Kirsh sees objects as extensions of our bodies, that we use to “think” with. Following this idea, we aim to understand how to design interaction with objects in VE in such way that this interaction changes user’s feelings, states, concepts, and beliefs. We look at the complex VE as a tool, the complexity of which enables for more interaction design opportunities. The potential of such tools is immense, and would allow for new research directions in embodied cognition and VR.

1.2 Unconventional interfaces

Conventional vs unconventional interfaces The definition of unconventional interfaces depends on the user’s familiarity with interaction interfaces. According to Kruijff [19], some of the characteristics of unconventional interfaces are: alternative input/output compared to hand-held control and audio-visual feedback, using either new technology or existing technology in a new way compared to conventional interfaces, use of interfaces in artistic works compared to common, everyday usages, and using “magical”/ unnatural metaphors compared to well-known metaphors.

As different tasks require fundamentally different interfaces (interfaces for a FitBit vs a flying interface in VR simulator), Beckhaus and Kruijff [4]:p.72 distinguish between interfaces used for “experimental application” and interfaces built with the goal of successful task-accomplishment (productive application). While in productive applications the main concern is usability, in experimental applications fun factor and aesthetics are some of the preferred values [4]:p.183. Here, we focus on the experimental application of a breath-controlled interface in our work. Given that there is no particular goal set for a user to achieve, but rather to explore the interaction, we argue that breathing as an interaction modality will contribute to a higher engagement of a user in VE, which will, we predict, enhance the user’s affective reactions to VE.

Breath-controlled Interfaces Respiration computer interfaces (RCI) are easy to use, accommodating to different body shapes, and preferred over button interfaces [2]. Two main categories of application for breath-controlled devices are:

assistive technologies for impaired individuals, and interfaces used for creative expression. Assistive technologies often take a form of breath controlled joysticks and mouses [20,13,29]. Breath-controlled interfaces in creative applications can be found in a number of video games [32,34] and in artworks that employ an audience's breath in various ranges of creative outputs [28,23].

The number of VR projects that have employed breathing input is limited, to our knowledge. Waterworth et al. [36] built a VE for exploring the relationship between emotion and presence, through a multimodal interaction paradigm. Employed input modalities were the user's balance for movement (leaning forward/backwards, right/left), and breath for vertical navigation. As the authors reported, initial trials confirmed the ease of using a breath interface for natural interaction on a vertical axis in VE. This project was inspired by the pioneering work of Char Daves' *Osmose*, an immersive VE presented on HMD, in which the user's breathing and balance were assessed via a vest [8]. This artwork, highly influential, refers to the phenomenological teachings of Heidegger and Merleau-Ponty in Daves' attempts to bring together divorced minds and bodies. In this work we can recognize traces of ideas of embodied interaction and cognition that will be publicly presented years after the completion of Daves' work.

1.3 Affect and Sound Estimation

Dimensional approach to affect estimation in sound and music: Affect estimation in sound and music is still in discussion in Music Information Retrieval (MIR) research. Eerola and Vuoskoski [10] argue for four affect models as discussed in their state of the art paper on affect estimation in sound and music [10]. The four affect models are: discrete, dimensional, miscellaneous, and music-specific. We undertake a dimensional approach to emotion that is focused on defining continuous dimensions that can represent and differentiate affective states. Ekkekakis [11] presented different dimensional models of affect in human subjects, however in our research we are focusing on the 2-dimensional circumplex model by Posner et al. [25]. The circumplex model has 2 axes, horizontal representing the valence dimension (unpleasant-pleasant) and vertical axis that represents the arousal or activation dimension (activation-deactivation), assessed here using the Affect Grid by Russell et al. [27]. We built *PBW* upon our previous research on affective estimation of soundscape recordings [14,5,35]. Specifically, we use the affect estimation model of Fan et al. [14] in our system design. In this 2-dimensional model of affect in audio, two axes are: pleasantness (equivalent to valence in human subject research) and eventfulness (equivalent to arousal).

Audio stimuli and Affective states: Previous research in the domain of audio stimuli and affective states showed that by varying pleasantness of the sound you can affect the user's ratings of arousal. Bradley and Lang showed that highly pleasant and highly unpleasant sounds had higher arousal ratings, while the most memorable sounds are those with high arousal [6]. Asutay and Västfjäll [3]

researched the relationship between emotional reactions as described through activation (arousal)-valence scales and the characteristics of sounds that are tones and noise complexes. Activation was related to the tonal content and sharpness, whereas valence was associated with perceived loudness, roughness, and naturalness. Similarly, Tajadura-Jiménez et al.[33] researched the determinants of sound properties (physical, physiological, and spatial) in regard to evoked affective responses and revealed the effect that the intensity of sound, amplitude and frequency modulation, and the type of sound (natural, and artificial) have on reported arousal. One of the relevant findings is that fast heartbeat sounds lead to increased reported arousal, and as well, can enhance affective response of the presented visual stimulus, if visuals and sounds are presented at the same time to a user. We are tackling this idea by matching changes in affective audio to the changes in visuals. While affect estimation in audio is a well researched area, there is no research done, to our knowledge, on the application of affect estimation of audio systems in VEs. Moreover, we research how changes in the audio and visual components of the VE triggered through embodied interaction influence the user's affect.

2 Pulse Breath Water, an immersive virtual environment with affect estimation in sound

Pulse Breath Water (*PBW*) is an immersive virtual environment (VE) presented to a user through a HMD and is manipulated by the pulse of a participant's breath, provoking and challenging the interaction between a user and the substantial element of the VE: water. The user rises in the VE when breathing in and slowly sinks (underwater) when breathing out. The interaction design follows the idea of "metaphoric" mapping (discussed in the Section 2.2 in more details). The audio is generated in real-time by mapping the eventfulness of the chosen audio samples to the frequency of the user's breathing.

Our design approach relied on an autobiographical design [24] and iterative research through design process [39]. The collaboration between the authors coming from HCI and generative audio field required iterative design sessions during which a variety of mappings between a user's breathing frequency, visuals, and audio were discussed and implemented.

2.1 Interaction Scenario

In the design process, we reduced visual impact following the concepts of ambiguity and abstraction. We decided to employ the user's breathing in an interaction with the VE, in a simple manner that empowered users to react to the system's decisions in VE through their breathing patterns. Users were comfortably seated, and we gave no instructions to the users prior to immersion; rather we left it up to them to explore the VE. The system recognizes subtle differences in breathing patterns, and reacts to changes in breathing patterns by changing the audio quality and visual characteristics (the waves become more calm as the breathing

slows down. e.g.). The system has its own behaviour that changes in regard to the incoming breathing patterns of a user. This process could be understood as negotiation between a user and the system, in a play that prioritizes the user's decisions over the system's.

2.2 System Description

The overall system outline is represented in Figure 1. Two breathing sensors (Thought Technology [1]) attached to the user's abdominal and chest area stream breathing waveform data to M+M middleware. M+M sends this data to a MAX msp patch. The reactive agent generates the audio output using an audio corpus (a set of pre-recorded audio samples). The reactive agent selects samples from its corpus using the mapping of the frequency of the user's abdominal breathing to the eventfulness of the audio samples. All audio samples were previously labelled with a two-dimensional vector: average eventfulness and average pleasantness using an affect estimation model proposed by Fan et al.[14]. The reactive agent sends online affect estimation of the audio output to Unity 3D along with breathing data via OSC messages. This data generates visual changes in the VE presented to the user via HMD. The user listens to the audio environment with circumaural noise-cancelling headphones.

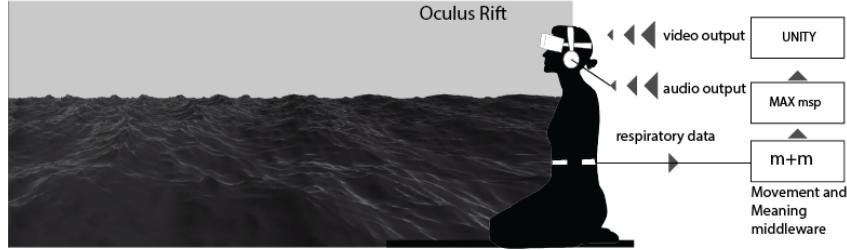


Fig. 1. The system architecture

Audio: Figure 2 shows the average affect values of each audio sample in *PBW*'s audio corpus. Each dot represents one audio sample. We created the audio corpus by recording two, three, four, and five voice chords with quartile harmony on the piano. Then, we used pitch shift and time stretch to generate more sounds. In particular, we used these methods to generate an audio corpus that locates around neutral valence and neutral to low arousal in the affective space. Following, we calculate the user's abdominal breathing patterns using the wavelet transform of the breathing data. In our implementation, the wavelet transform has 24 bands. We map these bands to the highest and the lowest arousal (eventfulness) values in our audio corpus. The reactive agent uses the band with the highest power to choose an audio sample. Hence, we map the

frequency of the user's breathing to the eventfulness of the audio. At any point, four audio samples are played together to ensure that the affective state of the overall audio centres around the neutral arousal. The design decision to position the audio corpus in this area of affective grid arose from authors' aesthetic tendencies. Our goal was to lead our users towards relaxing states, by introducing audio low in arousal (in audio vocabulary of affect: eventfulness), and staying in neutral to positive end of valence axis (pleasantness).

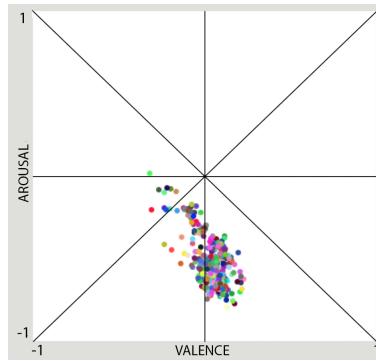


Fig. 2. The audio corpus of *PBW* mapped to 2 dimensional space defined by arousal and valence axes

Visual: Virtual environment built in Unity 3D comprises of a scene that combines interactive audio (generated independently via MAX msp patch) and the 3D element of a body of water - an ocean (see figure 3). The aesthetics of the scene is intentionally left minimal, displaying the ocean and the sky in a range of gray-scale shades over time (see figure 4). Below the main level of the ocean, we positioned an additional ocean surface in blue colour, to emphasize surrealism of the scene. A fog that encompasses the ocean in the distance adds to the ambiguity of the scene. We decided to implement these elements in order to maintain a neutral atmosphere dictated by neutral valence and arousal levels of the accompanying audio environment. This was based on the authors' judgment and several design iterations with informal user testing. The main design principle in designing this environment was ambiguity to evoke engagement and thought-provoking. As Gaver et al. [16] argue, ambiguity in HCI and design of interactive artifacts is desirable for the thought-provoking and engaging characteristics that it adds to the design. In *PBW*, we aimed at employing an "ambiguity of relationship" [16] that engages users to project their own values and experiences in the process of meaning-making. While meaning-making is not in the focus of the presented research, we find it to be a crucial component in creating the experience of the whole scene, adding to the affective potency of the environment.



Fig. 3. Screen shots of the environment: left: calm ocean; center: aroused ocean; right: under the water

Mappings: Breathing frequency as well as eventfulness (arousal) and pleasantness (valence) levels of the audio environment are sent from Max msp patch to the game engine Unity 3D. In Unity, the value of eventfulness is mapped to the waves of the ocean. Higher aroused states result in a more disturbed ocean surface and waves. The colour of the sky progresses from grey (at the beginning of the experience) to pitch black (at the end of a session) over the span of eighth minutes. A participant's breathing data controls the elevation of the user in the VE in that, when the user breathes in, their position in the environment is elevated so they can rise above the ocean surface. Similarly, when the participant exhales, they sink.

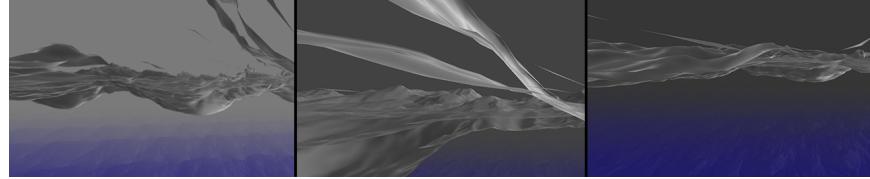


Fig. 4. Screen shots of the sky's colour progression: left: sky colour at minute 1; middle: sky colour at minute 5; right: sky colour at minute 8

2.3 PBW as an Art Installation

PBW was premiered as an art installation in two collective exhibitions: *Scores + Traces* at One Art Space in NYC, USA (March 10-12th, 2016), and at MUTEK-VR Salon in Montreal, Canada (November 9-13th, 2016). During these two exhibitions we gathered qualitative feedback from the audience, which we summarize here along with our observations of the audience's behaviour.

Easing-into the environment *PBW* was designed as a generative piece without a clear beginning or end. The time that the users spent in the *PBW* varied from 5 to 20 minutes. The users usually spend the first few minutes exploring the extremes of their breathing, to familiarize themselves with the system's capabilities through exaggerated belly movements while inhaling

and exhaling. Interestingly, after a few minutes of vigorous exploration, users would slowly ease-into the environment, and their breathing would become slowly paced. This type of breathing would typically remain stable until the end of each session (until the user decided they are done).

Meaning-making and re-evoked memories Even though we did not design *PBW* with any particular narrative in mind, the majority of users we spoke to had their own interpretation of what the narrative was. Some of them constructed the narrative, others re-lived some of their past experiences. We believe that a major role in the meaning-making process is held by users themselves, who invest their “beholder’s share” [17]. In other words, users respond to the ambiguity and lack of details by projecting their own experiences and imagination relying on top-down processing [17]:p.58.

3 Methodology

3.1 Study design

In order to investigate the potential benefits of predictable, embodied interaction through breathing on a user’s affect, enjoyment, engagement, and presence we designed an experimental comparison of interactions using two versions of *PBW* to support two different experimental conditions: (a) metaphoric mapping; and (b) reversed mapping;

3.2 Conditions and mappings

The original piece, *PBW*, was modified to support two experimental conditions that differed in interaction mapping between breathing frequencies and the changes in the environment.

Condition 1: Metaphoric mappings: In this condition, metaphoric mappings of audio and changes in VE are based on cognitive schema developed from everyday actions and interactions such as “more is up, less is down” [21]. Metaphoric mappings are widely exploited in the design of everyday objects (sliders moved up to “crank the volume up”) because the underlying concepts are understood beyond conscious awareness. For this reason metaphoric mappings of interactions are considered to be “intuitive” and require unconscious effort [21].

PBW was originally designed following the logic of metaphoric mappings. The vertical movement of the participant in the environment follows the logic of “more is up”: the more air you inhale the higher you move. When participants inhale they rise in the environments, and when they exhale they sink, similar to what happens when exhaling when swimming. The exact position is depending on the amount of air inhaled/exhaled, therefore the participant can be above the water (a big breath in), under the water (deep exhale), or any place in between if they maintain shallow breathing. In this metaphoric condition, we did not change the mapping between the respiratory interaction and the generative audio. We use the same mapping that we explain in Section 2.2 to generate the audio output.

Condition 2: Reverse mappings In this condition, we reversed the metaphoric mapping in order to investigate how this might affect participants' experiences in regard to their affect, engagement, immersion and overall satisfaction. In this condition, when a participant breathes in they sink, and rise when they exhale. This is a simple intervention yet clearly observable by the participants. The waves of the ocean were still mapped to arousal level. Moreover, we reversed the mapping between the respiratory sensor data and audio sample selection. As the user breathes more frequently, the reactive agent chooses samples with lower eventfulness; and vice versa.

Based on the above-mentioned cognitive schema and metaphor theory [21] we hypothesized that interaction based on *metaphoric mappings* will be more engaging and will enhance the affective properties of the audio more than the *reverse mapping* condition.

3.3 Participants

Twenty-four participants (16 female) were recruited using on-line participant recruitment system, and randomly assigned to one of the two experimental conditions to start with. Participants' ages ranged from 19 to 58 (mean: 22.3, SD: 8.03). Majority of participants have never tried VR before (14/24). All participants reported the good health condition and normal vision.

3.4 Experimental setup

The experiment was performed in iSpace lab, SIAT, SFU. The participants were seated, one at the time, in a dark room, at the computer station. Depending on theirs assigned experimental condition, one of the two VE experimental conditions were presented on an Oculus DK2 HMD (resolution 1080×960 per eye) and refresh rate of 75 FPS. The audio component of VE was played on noise cancelling headphones. Participants wore two breathing sensors (Thought technology) positioned on the abdomen and chest.

3.5 Procedure

Upon arrival in the lab, we informed participants that they are participating in an exploratory study in which we are interested in their engagement with the VE measured through assessed affect before and after, and additional questionnaires. Following, participants read written description of the study, and signed informed consent. The participants were informed about their rights to withdraw at any point and instructed to report to the experimenter any feelings of vertigo, nausea, or headache as they arise, upon which the experiment would be terminated. Each participant completed two eight minute long session (for example, condition 1: metaphoric mapping, and condition 2: reverse mapping). The order of conditions was counter-balanced across participants. After each session, the participants were interviewed.

3.6 Data collection

Before the experiment, the participants were asked to fill in the affect grid and state- trait anxiety inventory -STAI-6 [22]. After each exposure, the participants filled in the affect grid and STAI-6 again, without seeing their previous responses. In addition, they were asked to answer a questionnaire containing twenty-one questions. Our questionnaire is a modified version of the Game Engagement Questionnaire [7] used for assessing levels of engagement through the lenses of four categories: flow, immersion, engagement, and presence. Following, the participants were interviewed and the interviews were audio recorded.

3.7 Data analysis

Data analysis was performed on the data from twenty-two participants. Data from two participants had to be discarded: One participant experienced anxious feeling in the middle of the first exposure and the experiment was stopped at that point. The other participant did not report motion sickness as it occurred and rather continued, but was unable to complete all of the questionnaires. Quantitative data was analyzed through inferential statistics, as explained below in the Findings section. Interviews were transcribed and analyzed using a grounded theory approach. The deductive approach to coding originated from the semi-structured interview questions that focused on the experience: feelings, thoughts, actions performed, attention, intentions, narrative, evoked memories, and difficulties of using the system.

4 Quantitative findings

4.1 Questionnaire findings

A two-way within-subject ANOVA was run on a sample of 22 participants to examine the effect of order and mapping on the different questionnaire items. Below we only report significant main effects and interactions.

Perceived reactivity of the environment to the user: There was a significant main effect of order, in that participants perceived the environment as more reactive in their second exposure, $F(1, 40) = 2.95, p = .013$, as illustrated in Figure 5 right.

Users engagement to change the sounds and visuals: Participants purposefully used their breath to manipulate the environment in their second exposure more than in their first exposure $F(1, 40) = 2.20, p = .016$ (see Figure 5 left).

Payed attention to the audio: Participants payed more attention to the audio in metaphoric as compared to the reverse mapping condition $F(1,40) = 1.76, p = 0.039$ (see figure 6 left).

Desire for experience to last longer: There was a significant interaction between order and mapping for the questionnaire item "I wish it lasted longer", $F(1, 39) = 6.14, p = .0177$ (see Figure 6 right). Planned contrasts

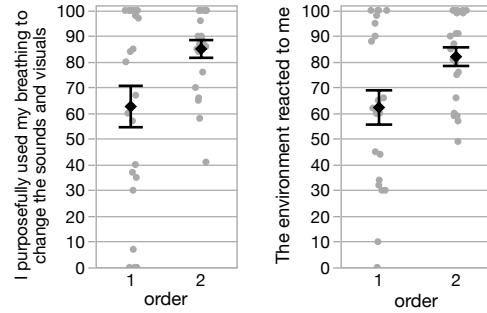


Fig. 5. Main effect of order on the questionnaire dependent variables "I purposefully used my breath to change the sounds and visuals" (left) and "The environment reacted to me" (right). Error bars depict one standard error of the mean. Grey dots depict individual participants' mean values.

showed that after the second session participants were more inclined to wish for a longer experience if this second experience was the metaphorically mapped condition versus the reverse mapping condition, $F(1, 39) = 5.56, p = .0233$. If the metaphoric condition was experienced as the second session, participants were also more inclined to wish for a longer experience than if the metaphoric condition was experienced first, $F(1, 39) = 5.22, p = .0278$.

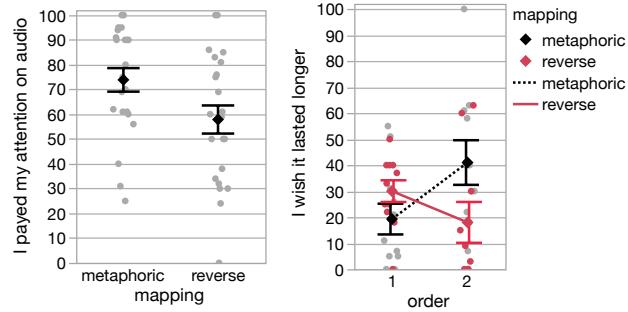


Fig. 6. Main effect of mapping on the questionnaire dependent variable "I payed my attention on audio" (left) and interaction between order and mapping on "I wish it lasted longer" (right). Error bars depict one standard error of the mean. Individual dots depict individual participants' mean values.

4.2 Affect Grid and STAI-6

A 2-way ANOVA for the factors order {baseline before the first session; after session 1; after session 2 } and mapping {metaphorical; reverse} and the dependent variables arousal and pleasantness scores from the affect grid did not show any significant main effects or interactions. In regard to the six questions included in STAI-6, we found a significant difference in baseline (pre-exposure) scores between the two groups (one group that was assigned to metaphoric mapping condition first, and the other one that started with the reverse condition), even though participants were randomly assigned to the two groups. Due to these group differences we did not further analyze the STAI-6 results.

5 Interviews

The majority of the participants in our study were undergrad students with no prior exposure to virtual reality. Through semi-structured interviews after each condition we hoped to gather insights that will help us build a better understanding of how different interaction mapping contributes to the affective properties of the environment, and overall experience regardless of the previous experiences with the technology. The themes from semi-structured interviews served as a basis for the non-linear accounts of various experiences as presented here.

5.1 Exploring the unknown: phases

The majority of the participants verbally shared their excitement to try VR for the first time. As we noticed, the first phase of interaction is exploration of their agency by breathing in and out, testing the limits of the system (how high or low they can get), and familiarizing themselves with the elements in the environment. After this exploration phase, they eased into the environment.

“In the beginning I breathe in different levels so I can see how the image will move. After I realized how it works, I tried different kinds of breathing. I tried even to go forward but I couldn’t. Oh, at the beginning I was a little bit worried that image will be intimidating, but after I realized it was ocean it was more relaxed. Then I tried different kind of stuff that didn’t work so I kept breathing”. [P9]

5.2 Regardless of mapping, second trial is enjoyed more

The participants reported second trials as more enjoyable very consistently, regardless of the mapping. Even though participants were informed how the system works, they used their first trial to familiarize themselves with the environment to allow for more profound interaction in the second trial.

“This is my first time to try VR, in the first environment I felt... don’t want to say stressed... maybe anxious a little bit, a little bit excited, the second time my perception was: ok, this is stuff I already know, it’s like an old friend, I know what to

expect, I know what should I do, observe... I enjoyed it more the second time. [P1]
 The lack of anticipation of a new environment one is immersed in resulted in increased relaxation in the second exposure.

"The first time was like giving a toy to a child... this time I was enjoying the feeling of calm... I wanted to take good relaxation time now". [P2]

"The first time I was not sure what you would ask me, or what to expect... this time I knew what was coming... There were some parts that were intense... but I was immersed in the simulation... I knew nothing crazy is going to happen, I was more calm" [P14].

5.3 Metaphoric mapping feels intuitive, but reverse is more playful

Those participants who were aware of the differences in interaction mapping between two conditions articulated their preferences for metaphoric mapping. The descriptions of the metaphoric mapping conditions such as: intuitive, natural, and counter-intuitive emerged from participants' comments, and were usually linked to the themes such as relaxation, and being calm.

"I practised being at the water level, tried going down below blue waves, but you need a big breath for that. The other one [metaphoric] felt more natural, I guess, because you breathe in and go up, breathe out and go down, and this one felt weird because it is opposite" [P5].

"I feel much better than in the first one (referring to reverse mapping)... it felt so correct, when I am in the water you exhale and go down... I felt more calmer than when I came in" [P2].

On the contrary, reverse mapping was perceived as more stimulating, engaging, and interesting.

"They are both good for different (reasons): reflected more in the first one (metaphoric), in this one I was more playful (reverse)" [P5].

"Not much different but felt more interesting, it was counter-intuitive" [P7]

5.4 Somatic experiencing: awareness of breath and emerging past experiences through the changes in the environment

Visual representation of the ocean coupled with the movement often triggered memories in participants, followed by strong bodily sensations such as: floating, dropping, or even sensations that "...I could almost feel, like being submerged under the water and then being brought back up... like if there was an invisible wrapper, kind of like a real water but not as... kind of real water but at slower pace." [P14]

"It was exactly like when you go in the water, it had the feeling of being calm... it felt like floating chamber... the idea is the same, idea is that water makes you feel calm, floating in the water" [P2]

"When I was learning to swim... they tell you to focus on your movement, and breathing... I related that to this. What made me feel different was motion... I liked the first one (reverse), it made me feel better... the second one (metaphoric) felt more like I was floating" [P4].

Few participants reported a heightened impact that the music had on their awareness of breath.

"I think it was music. Suddenly I felt like I can feel my heartbeat. The rhythm of the music... it was in the parallel with my breathing, and that's when I noticed (her breath) [P17].

“Oh, when the music gets excited I know something might happen, and then I wait for it happen. Oh, I also breathe to let it happen more faster” [P9]. However, some participants experienced tension in regard to audio they were listening to:

“I felt a heartbeat and then when it was music every time I went down... I felt dramatic effect to it... so I was like: something is wrong, I should do something about it... it was triggering fear even though I knew it’s VR... when the sky turned black, sounds triggered fear” [P3].

5.5 Loss of control triggers fear?

From the conversation with the participants we realized that those participants who did not make a connection between their breath and the changes in the environment were more likely to get distressed by audio or visuals.

“I tried to control my breathing... and then once the loud music started I kind of... can’t control it any more, and I felt tense... I lost control of how I wanted to go down to the blue, and that’s while loud music started to fade and I regained control... then I started to take deep breaths and started to calm myself...” [P21]

Once the participants regained control over the system, their tension lessens.

“Fear doesn’t come to me this time, I felt I have a control over my body... to stay in one state, tried to be at one situation... either above or down” (the second exposure, metaphoric mapping) [P2].

5.6 Imagine this was a tool...

During the interview, the participants were asked “If I tell you that what you just experienced is a tool, what would you use that tool for?”. Two themes emerged as the most dominant: a relaxation tool, and a tool for overcoming anxieties triggered by water.

“I would use it for, like, calming people down... cause this probably can calm many people down who are stressing about stuff... it gives you something to focus on, you are focusing on your breathing and something in front of you, so it kind of distracts you from everything else... cause you think of, if you start panicking, I guess, and you breathe really fast and go up and down like crazy, and then you would be like: what’s happening in the world, and then your focus immediately is on the image in front of you instead anything outside.” [P6].

“I don’t know... something to do with calming people down, when their heads are somewhere else” [P5]

“Zoning out, not thinking about whatever is going in your head... If things were steady up the water... I would feel more relaxed.” [P2].

6 Discussion

In this section we discuss the main findings from the presented study and our understanding of the lived experiences of the participants. We asked the question: *how can these environments change us through embodied interaction design? And, how can an embodied interaction design that employs the user’s subtle breathing movements facilitate these changes?* and the answer lies between multiple accounts gathered here. The

richness of gained insights helped us to see a wide range of factors that can affect the experience, and that we did not take into the account during the study planning process. Finally, we discuss the insights as we formalize them into a set of design considerations for embodied interaction design in VR.

6.1 Familiarity first, engagement after

Analysis of questionnaire responses highlighted that the participants perceived the environment as more reactive to their input in their second trials, regardless of the mapping. We believe that this can be explained by the novelty effect or lack of understanding of the system's nuances. Even though prior to each trial we explained how the system works, many of the participants did not make any connection between their breathing and the changes in the environment in their first trial. This would explain our second finding that the engagement of the user to manipulate the environment through their breathing was higher in the second trial as well. Once they knew how the system worked they engaged with it more. The dynamic of their familiarization was revealed in the interviews in which the majority of the participants revealed that in the beginning they were exploring the environment and testing the interaction limits, followed by easing into it and pacing their breathing in less forceful and more pleasurable ways. Two participants were not aware of their agency at all. One reason might be that these participants employ chest breathing more than abdominal breathing, which will be explored in future work.

6.2 Tension and relaxation, at the same time

We investigated whether different mappings can lead participants' affect toward an affect that matches the overall affect of the audio corpus. The audio corpus was intentionally centred around neutral pleasantness with a tendency towards positive pleasantness and neutral to low arousal (playfulness), positioning the corpus in the area of relaxed feelings on the affect grid. The inferential data analysis of affect grid responses did not yield any significant differences between two different mappings nor trial order and the baseline. This might be explained by overall subtle changes in the affect across the sample. However, a few participants reported feelings of tension. From the interviews we learned that many of them found the sky colour change from grey to black dreadful and this element triggered anxious-like feelings in several participants. One participant finished the session after four minutes claiming that the environment caused her distress and she was not able to continue. Other participants who did not make the connection between their agency and the changes felt tensed too. In these cases, the music was adding tension to the dark environment. Despite the reports of felt tension, when asked what they would use this tool for, the majority of the participants responded that they would use it for relaxation. Even though there were elements that were causing distress, the participants recognized calming qualities of the system. This finding is of a particular interest to our future work.

6.3 The context is the key

Originally, we designed *PBW* as an art piece grounded in research questions we asked here. As an art piece, we exhibited it in galleries where we verbally collected the experiences of audience members who interacted with the piece. The majority of them

recognized relaxing qualities and would stay immersed up to 20 minutes. The quality of the experience dramatically changed when we moved the setup from the gallery to the lab. The main difference was the expectation and the openness to the experience. The audience in the gallery is there because they would like to experience something new. Our participant pool consisted of fairly young undergrad students who might be very different from those who initially experienced the piece in an artist gallery context. The laboratory setting, no matter how we tried, still feels like the setting for an experiment rather than an experience. This might have affected our participants' responses and we find it to be an important factor to be accounted in the future studies that employ art and research questions.

7 Conclusion

In this paper, we introduced the system *Pulse Breath Water* and we investigated the efficiency of embodied interaction design through two different mappings (metaphoric, and “reverse”) for enhancing affective properties of the system. This research encompasses two directions of affective research: in VR and in the audio, combining them into one system in an attempt to gain a better understanding of the combined effect of these two on the user’s engagement, affective states, immersion, and overall experience.

In this paper we contribute to a better articulation of affective properties of virtual environments that combine visual and audio components into one system. We presented some of the individual accounts of lived experiences and showed that the majority of the participants when asked to imagine that this was a tool replied that they would use this tool for relaxation. We built our system on the premise of neutral pleasantness and low arousal properties which can be translated to feelings of being relaxed. This gives us a direction for future work to research the potential of the system of inducing a wider range of affective states. We believe that the insights presented here will bring us closer to the final goal of creating a system that not only “reacts” to a user’s breathing but evolves into an immersive artificial intelligence system capable of taking initiative and changing a user’s affective states.

8 Acknowledgements

We thank all the study participants for their involvement, and the *MovingStories* SSHRC research project for their support while working on this piece.

References

1. Thought Technology Ltd. ProComp2 - 2 Channel Biofeedback & Neurofeedback System w/ BioGraph Infiniti Software Thought Technology Ltd., Apr. 2015. <http://thoughttechnology.com/index.php/procomp2-2-channel-biofeedback-neurofeedback-system-w-biograph-infiniti-software.html>.
2. ARROYO-PALACIOS, J., AND ROMANO, D. M. Exploring the use of a respiratory-computer interface for game interaction. In *2009 International IEEE Consumer Electronics Society’s Games Innovations Conference* (Aug. 2009), pp. 154–159.
3. ASUTAY, E., AND VÄSTFJÄLL, D. Perception of Loudness Is Influenced by Emotion. *PLoS ONE* 7, 6 (June 2012).

4. BECKHAUS, S., AND KRUIJFF, E. Unconventional Human Computer Interfaces. In *ACM SIGGRAPH 2004 Course Notes* (New York, NY, USA, 2004), SIGGRAPH '04, ACM.
5. BERGLUND, B., NILSSON, M. E., AND AXELSSON, Å. Soundscape psychophysics in place. IN07-114.
6. BRADLEY, M. M., AND LANG, P. J. Affective reactions to acoustic stimuli. *Psychophysiology* 37, 2 (Mar. 2000), 204–215.
7. BROCKMYER, J. H., FOX, C. M., CURTISS, K. A., MCBROOM, E., BURKHART, K. M., AND PIDRUZNY, J. N. The development of the Game Engagement Questionnaire: A measure of engagement in video game-playing. *Journal of Experimental Social Psychology* 45, 4 (July 2009), 624–634.
8. DAVIES, C. OSMOSE: Notes on being in Immersive virtual space. *Digital Creativity* 9, 2 (Jan. 1998), 65–74.
9. DOURISH, P. *Where the Action Is*. MIT Press, 2004.
10. EEROLA, T., AND VUOSKOSKI, J. K. A review of music and emotion studies: approaches, emotion models, and stimuli. *Music Perception: An Interdisciplinary Journal* 30, 3 (2013), 307–340.
11. EKKEKAKIS, P. Should affective states be considered as distinct entities or as positioned along dimensions? In *The Measurement of Affect, Mood, and Emotion: A Guide for Health-Behavioral Research*. Cambridge University Press, Feb. 2013, pp. 52–72.
12. ENGLAND, D., RANDLES, M., FERGUS, P., AND TALEB-BENDIAB, A. Towards an advanced framework for whole body interaction. In *International Conference on Virtual and Mixed Reality* (2009), Springer, pp. 32–40.
13. EVREINOV, G., AND EVREINOVA, T. "Breath-Joystick"-graphical manipulator for physically disabled users. *Proc. of the ICCHP2000*, 193 200 (2000).
14. FAN, J., THOROGOOD, M., AND PASQUIER, P. Automatic soundscape affect recognition using a dimensional approach. *Journal of the Audio Engineering Society* 64, 9 (2016), 646–653.
15. FLACH, J. M., AND HOLDEN, J. G. The Reality of Experience: Gibson's Way. *Presence: Teleoperators and Virtual Environments* 7, 1 (Feb. 1998), 90–95.
16. GAVER, W. W., BEAVER, J., AND BENFORD, S. Ambiguity As a Resource for Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2003), CHI '03, ACM, pp. 233–240.
17. KANDEL, E. *Reductionism in Art and Brain Science: Bridging the Two Cultures*. Columbia University Press, Aug. 2016.
18. KIRSH, D. Embodied Cognition and the Magical Future of Interaction Design. *ACM Trans. Comput.-Hum. Interact.* 20, 1 (Apr. 2013), 3:1–3:30.
19. KRUIJFF, ERNST. *Unconventional 3D User Interfaces for Virtual Environments*. Doctoral Dissertation, Oct. 2006.
20. KUZUME, K. Input device for disabled persons using expiration and tooth-touch sound signals. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (2010), ACM, pp. 1159–1164.
21. MACARANAS, A., ANTLE, A. N., AND RIECKE, B. E. What is intuitive interaction? balancing users' performance and satisfaction with natural user interfaces. *Interacting with Computers* 27, 3 (2015), 357–370.
22. MARTEAU, T. M., AND BEKKER, H. The development of a six-item short-form of the state scale of the Spielberger State-Trait Anxiety Inventory (STAII). *British Journal of Clinical Psychology* 31, 3 (1992), 301–306.
23. NET, M. A. Media Art Net | Gabriel, Ulrike: Breath, 1992/93, Dec. 2016.

24. NEUSTAEDTER, C., AND SENGERS, P. Autobiographical design in hci research: designing and learning through use-it-yourself. In *Proceedings of the Designing Interactive Systems Conference* (2012), ACM, pp. 514–523.
25. POSNER, J., RUSSELL, J. A., AND PETERSON, B. S. The circumplex model of affect: An integrative approach to affective neuroscience, cognitive development, and psychopathology. *Development and Psychopathology* 17, 3 (2005), 715–734.
26. ROBERTSON, T. Cooperative Work and Lived Cognition: A Taxonomy of Embodied Actions. In *Proceedings of the Fifth European Conference on Computer Supported Cooperative Work*. Springer Netherlands, 1997, pp. 205–220. DOI: 10.1007/978-94-015-7372-6_14.
27. RUSSELL, J., WEISS, A., AND MENDELSOHN, G. Affect Grid - a Single-Item Scale of Pleasure and Arousal. *Journal of Personality and Social Psychology* 57, 3 (Sept. 1989), 493–502.
28. SCHIPHORST, T. Breath, skin and clothing: Using wearable technologies as an interface into ourselves. *International Journal of Performance Arts and Digital Media* 2, 2 (2006), 171–186.
29. SHORROCK, T. H., MACKAY, D. J. C., AND BALL, C. J. Efficient Communication by Breathing. In *Deterministic and Statistical Methods in Machine Learning*. Springer, Berlin, Heidelberg, 2005, pp. 88–97. DOI: 10.1007/11559887_5.
30. SLATER, M. Place illusion and plausibility can lead to realistic behaviour in immersive virtual environments. *Philosophical Transactions of the Royal Society of London B: Biological Sciences* 364, 1535 (Dec. 2009), 3549–3557.
31. SLATER, M., AND SANCHEZ-VIVES, M. V. Enhancing Our Lives with Immersive Virtual Reality. *Frontiers in Robotics and AI* 3 (2016).
32. SONNE, T., AND JENSEN, M. M. Chillfish: A respiration game for children with adhd. In *Proceedings of the TEI'16: Tenth International Conference on Tangible, Embedded, and Embodied Interaction* (2016), ACM, pp. 271–278.
33. TAJADURA-JIMENEZ, A., VÄLJAMÄE, A., AND VÄSTFJÄLL, D. Self-representation in mediated environments: the experience of emotions modulated by auditory-vibrotactile heartbeat. *Cyberpsychology & Behavior: The Impact of the Internet, Multimedia and Virtual Reality on Behavior and Society* 11, 1 (Feb. 2008), 33–38.
34. TENNENT, P., ROWLAND, D., MARSHALL, J., EGGLESTONE, S. R., HARRISON, A., JAIME, Z., WALKER, B., AND BENFORD, S. Breathalising games: understanding the potential of breath control in game interfaces. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology* (2011), ACM, p. 58.
35. THOROGOOD, M., AND PASQUIER, P. Impress: A Machine Learning Approach to Soundscape Affect Classification. pp. 256–260.
36. WATERWORTH, E. L., HÄGGKVIST, M., JALKANEN, K., OLSSON, S., WATERWORTH, J. A., AND WIMELIUS, H. The exploratorium: An environment to explore your feelings. *PsychNology Journal* 1, 3 (2003), 189–201.
37. WILSON, S. *Information arts: intersections of art, science, and technology*. Leonardo. MIT Press, Cambridge, Mass, 2002.
38. ZAHORIK, P., AND JENISON, R. L. Presence as Being-in-the-World. *Presence* 7, 1 (Feb. 1998), 78–89.
39. ZIMMERMAN, J., FORLIZZI, J., AND EVENSON, S. Research through design as a method for interaction design research in hci. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (2007), ACM, pp. 493–502.