CMMR

2021TOKYO

The 15th International Symposium on
Computer Music Multidisciplinary Research

Music in the AI Era

ONLINE, 15 - 19 November 2021

Proceedings of the

15th International Symposium on Computer Music Multidisciplinary Research

15 – 19th November, 2021
Online

CMMR 2021 Organizing Committee, Japan

in collaboration with

The Laboratory PRISM
"Perception, Representations, Image, Sound, Music"
Marseille, France

PRISM

Published by

CMMR 2021 Organizing Committee, Japan

in collaboration with

The Laboratory PRISM
“Perception, Representations, Image, Sound, Music”
Marseille, France

November, 2021

Proceedings Editors: T. Kitahara, M. Aramaki,
R. Kronland-Martinet, S. Ystad

ISBN 979-10-97-498-02-3
Les éditions de PRISM

Welcome to CMMR 2021

We are pleased to welcome you to the 15th edition of CMMR being held online. We hope that by participating in CMMR 2021 and actively interacting with each other, you will be able to actively exchange ideas, gain rich inspiration, and make great progress in your research.

The global corona disaster also had a great impact on the CMMR conference. Originally, we, the organizing committee, had planned to hold the CMMR in Tokyo in November 2020. The deadline for submissions was set at April 2020, and the call for papers was distributed. However, since we could not foresee the end of COVID-19 infection unfortunately, we made a tough decision to postpone the conference for one year, only two weeks before the deadline of April. Thus, we are finally very happy to be able to hold the conference online and to welcome so many participants!

CMMR 2021 takes place using Zoom, Slack, and YouTube over a period of five days, and the sessions have been arranged according to Asian, European, and American time zones. To take advantage of the nature of the online conference, we have made the participation fee free for those who only watch the sessions. Instead of giving up social events such as receptions and banquets, we have set up a number of interaction channels on Slack, and in addition to Zoom, we also provide simultaneous and archived streaming on YouTubeLive.

The conference theme of CMMR 2021 has been set at "Music in the AI Era." In music informatics, interdisciplinary collaborative research has already been conducted with various related fields such as linguistics, brain science, psychology, sociology, pedagogy, and art. Recently, the rapid development of artificial intelligence technology is changing not only the nature of interdisciplinary collaborative research, but also the meaning of music for people and society. CMMR 2021 aims to share knowledge with participants who share a common background in music informatics, through in-depth discussions on the current status and vitalization of interdisciplinary research and the use of artificial intelligence technology.

We are delighted to include a keynote speaker and two invited speakers based on the conference theme "Music in the AI Era." The keynote lecture is given by Prof. Shuji Hashimoto (Professor Emeritus and former Vice President of Waseda University, and former Vice President of the International Computer Music Association), and the invited lectures are given by Dr. Gaetan Hadjeres (SONY CSL, Paris) and Prof. Tadahiro Taniguchi (Professor, Ritsumeikan University). The music works are available on YouTube, and we have set up the music sessions in which composers explain music works.

Lastly, I briefly introduce the organizing committee. In Japan, the Special Interest Group on Music (SIGMUS) has been active since 1993, and playing a key role in

incubating music informatics in collaboration of industry and academia. Most of the members of the organizing committee belong to SIGMUS, and have been engaged in research activities for a long time. SIGMUS financially supports CMMR 2021.

We would like to express my sincere gratitude to all the members the Scientific Program Committee, Music Committee, and Steering Committee, and the sponsors for their cooperation in organizing CMMR 2021.

Keiji Hirata
General Chair

Message from Scientific Program Chairs

Thank you for attending the 15th International Symposium on Computer Music Multidisciplinary Research (CMMR 2021), the first online conference in a series of CMMR conferences. Holding CMMR 2021 in a fully online format was a very difficult decision for us. When we decided to postpone CMMR 2020 for one year, we expected that we could hold the conference onsite in 2021. However, the worldwide COVID-19 pandemic has not yet ended; therefore, we decided to hold CMMR 2021 online.

Despite the online conference, the paper review process was carried out in almost the same way as for the past CMMR conferences. Each submission was peer reviewed by three experts in principle. The review process was single-blind. From various regions in the world, including Japan, Europe, the United States, Canada, Brazil, and India, 48 papers were submitted. Out of them, 33 papers were accepted. As the conference name suggests, these papers cover a wide range of topics including audio signal processing, music information retrieval, artistic applications of artificial intelligence, performance modeling, and computational music analysis. Although recent CMMRs included poster and/or demo presentations, we collected only long and short papers (10 and 6 pages, respectively) with oral presentations to make the online conference as simple as possible.

The most important policy in organizing CMMR 2021 in the online format is to encourage both synchronous and asynchronous discussions. To encourage synchronous discussions, we will use Zoom, an online video meeting platform. Each presenter will present his/her work on Zoom using the screen share function. In addition, we will set up an opportunity for discussions, namely, *post-session discussions*, using Zoom's breakout room function after each session. We sometimes enjoy discussion at the coffee breaks in onsite conferences. The post-session discussion aims to provide a similar opportunity to do this. To encourage asynchronous discussions, we use YouTube and Slack. Each presentation will be broadcast on YouTube Live and archived as YouTube video content. Participants can therefore watch presentations after the sessions. We think this process will enable people worldwide to more easily participate in the conference because some sessions will be held at midnight in some time zones. After watching video presentations on YouTube, the participants will be able to ask questions on our Slack workspace. To encourage discussions on Slack, we have established a separate channel for each presentation.

The music program, which is also an important part of CMMRs, was also affected by COVID-19. When we planned to hold the conference onsite, we were preparing a live concert at a hall in Japan. Unfortunately, however, the music program committee (Chair: Prof. Shintaro Imai) had to decide to change a place for presenting musical works from a real live concert to online video sharing on YouTube. Nevertheless, 32 various musical works from over the world were submitted, and 13 were accepted.

We would like to thank all the people who submitted and reviewed papers and all the participants of the conference. We would, in particular, like to thank the program committee members who had to review many papers within only one month, because the paper deadline was set one month later than usual to encourage a large number of paper submissions. Without each program committee member's cooperation, it would not have been possible to hold the conference.

We hope all participants enjoy the conference.

On behalf of the scientific program committee chairs,
Tetsuro Kitahara

Organization

General Chair

Keiji Hirata (Future University Hakodate, Japan)

General Co-Chair

Satoshi Tojo (JAIST, Japan)

Scientific Program Chairs

Tetsuro Kitahara (Nihon University, Japan)
Mitsuko Aramaki (AMU-CNRS-PRISM, France)
Richard Kronland-Martinet (AMU-CNRS-PRISM, France)
Sølvi Ystad (AMU-CNRS-PRISM, France)

Music Chair

Shintaro Imai (Kunitachi College of Music, Japan)

Website

Masatoshi Hamanaka (RIKEN, Japan)

Treasurer

Masaki Matsubara (University of Tsukuba, Japan)

Proceedings Chair/Registration Chair

Aiko Uemura (Nihon University, Japan)

Local Organizer

Hidefumi Ohmura (Tokyo University of Science, Japan)
Ryo Hatano (Tokyo University of Science, Japan)
Shun Sawada (Tokyo University of Science, Japan)

Secretary

Nao Aoki (JAIST, Japan)

Scientific Program Committee

Tetsuro Kitahara (Nihon University, Japan)
Mitsuko Aramaki (AMU-CNRS-PRISM, France)
Richard Kronland-Martinet (AMU-CNRS-PRISM, France)
Sølvi Ystad (AMU-CNRS-PRISM, France)
Gilberto Bernardes (INESC TEC, Portugal)
Tifanie Bouchara (CNAM, France)
Marcelo Caetano (CIRMMT-McGill, Canada)
F. Amílcar Cardoso (University of Coimbra, Portugal)
Roger Dannenberg (Carnegie Mellon University, USA)
Matthew Davies (University of Coimbra, Portugal)
Georg Essl (University of Wisconsin – Milwaukee, USA)
Satoru Fukayama (AIST, Japan)
Masatoshi Hamanaka (RIKEN, Japan)
Tatsunori Hirai (Komazawa University, Japan)
Keiji Hirata (Future University Hakodate, Japan)
Katsutoshi Itoyama (Tokyo Institute of Technology, Japan)
Sven-Amin Lembke (De Montfort University, UK)
Luca Andrea Ludovico (University of Milan, Italy)
Sylvain Marchand (University of La Rochelle, France)
Eita Nakamura (Kyoto University, Japan)
Marco Buongiorno Nardelli (University of North Texas, USA)
Charalampos Saitis (Queen Mary University of London, UK)
Charlesde Paiva Santana (IRCAM, France)
Diemo Schwarz (Ircam - CNRS STMS, France)
Bob Sturm (KTH, Sweden)
Etienne Thoret (Schulish School of Music, McGill University, Canada)
Satoshi Tojo (JAIST, Japan)
Adrien Vidal (AMU-CNRS-PRISM, France)
Ryosuke Yamanishi (Kansai University, Japan)

Music Committee

Shintaro Imai (Kunitachi College of Music)
Kiyoshi Furukawa (Tokyo University of the Arts)
Johnathan F. Lee (Tamagawa University)
Haruka Hirayama (Hokkaido Information University)
Asako Miyaki (Shobi University)

Steering Committee

Mitsuko Aramaki (PRISM, AMU-CNRS, France)
Mathieu Barthet (QMUL, UK)
Matthew Davies (INESC TEC, Portugal)
Richard Kronland-Martinet (PRISM, AMU-CNRS, France)
Sølvi Ystad (PRISM, AMU-CNRS, France)

Sponsors

YAMAHA

SIGMUS
IPSJ SPECIAL INTEREST GROUP
ON MUSIC AND COMPUTER
音楽情報科学研究会

公益財団法人栢森情報科学振興財団
Kayamori Foundation of Informational Science Advancement

Tateisi
Science and Technology
Foundation

Table of Contents

Audio Signal Processing

Music Analysis 1

Music Analysis 2

Music Information Retrieval and Modeling 2

Audio Signal Processing and Performance Modeling

AI, ML, and Electroacoustics for Music Production

Music Database and Ontology

Keynote Lecture

Shuji Hashimoto

Professor Emeritus and Research Advisor, Waseda University

Shuji Hashimoto received the B.S., M.S. and Dr. Eng. Degrees in Applied Physics from Waseda University, Tokyo, Japan, in 1970, 1973, and 1977, respectively. He was an Associate Professor in the Department of Physics, Toho University from 1979. In 1991 he moved to Waseda University as a Professor of the Department of Applied Physics. In Waseda University he served as the Director of the Humanoid Robotics Institute for ten years from 2000. During 2006-2010 he was the Dean of Faculty of Science and Engineering. He was appointed and served as the Senior Executive Vice President for Academic Affairs and Provost of the University from 2010 to 2018. He has been one of the leaders of the Gundam Global Challenge since 2014. Currently he is a Professor Emeritus and Research Advisor of Waseda University. He joined XELA Robotics as the CEO in April, 2019. His research interests include Artificial Intelligence, Robotics, "KANSEI" Information Processing, Sound and Image Processing and Meta-Algorithm.

Lecture: Music in the AI era

Many times, we were declared "Finally real artificial intelligence has been completed" and we were betrayed each time with various excuses. However, looking at the recent progress in AI technology, it seems that this time it might be true. Powerfully connected computers with big data seem to present adequate solutions to complicated problems that could not be solved ever before.

Science and engineering have been an integral inseparable to form technology. Science organizes discovered knowledges and construct the theory to understand, while engineering presents means and methods that put the theory into practical use to provide solutions to real world problems. But presently the deep-learning-based AI produces solutions directly from a huge accumulation of raw data. It seems that science is blown off from the traditional picture of technology. The rest is engineering alone that delivers solution. At present, AI works well most but not all. However, it does not tell us why the answer is correct. As many people complain, there is no proof of validity. The output of AI often sounds like God's revelation. It is a black box we never know its inside. What we can do is only to believe in AI, saying that "because the computer is aware of all." With the recent rise of AI, traditional decent researchers, who accumulate appropriate processes based on theory and knowledge to approach the solution, seem

to have been exiled from the main stage in many fields including music technology and science.

Science seems to be at stake in this way, but I am not pessimistic about the current situation. We need a science to understand things. We need engineering to make things. Science hates black box. While engineering often accept black box if it is useful. Useful tools accelerate science. AI is not yet in the final stage neither human intelligence is not. I believe we needs to start a new story of science together with a new tool AI. Music is fascinating field in elucidating human intelligence and creativity as it contains philosophy and arts, science and engineering, I would like to talk my story on Music in the AI Era.

Invited Lectures

Gaëtan Hadjeres
Sony CSL Paris Music Team

Gaëtan Hadjeres graduated from the École Polytechnique (France) and obtained a master in Pure Mathematics from Paris 6 University (Sorbonne Universités). He joined Sony CSL Paris in 2014 to do a Ph.D. thesis on music generation under the supervision of François Pachet and Frank Nielsen. In 2018, Gaëtan successfully defended his dissertation entitled "Interactive Deep Generative Models for Symbolic Music" and is now a permanent member of the Sony CSL Paris Music Team. Parallel to his scientific background, he studied music composition at the Conservatoire de Paris (CNSMDP) and he is also a pianist and a double bass player. His works (DeepBach, the Piano Inpainting Application) focus on the creation of A.I. tools able to assist musicians during composition, enrich their creative process and make music composition playful and accessible to a wide audience.

Lecture: Developing Artist-centric Technology

Important progress in generative modeling has been made over the last few years, allowing researchers to envision novel creative usages with impressive results. However, we can notice that such A.I. algorithms are often not easily accessible or controllable by an artist, so that their widespread adoption by content creators is yet to come. In this talk, I will present various examples of our modular approach at Sony CSL to bridge the gap between researchers and artists through the development of A.I. assistants. Setting the interaction with an artist as our core requirement brings up new interesting challenges and we hope it will help democratizing the latest advances in A.I. amongst musicians.

Tadahiro Taniguchi

Professor, College of Information Science and Engineering,
Ritsumeikan University

Tadahiro Taniguchi received the ME and Ph.D. degrees from Kyoto University in 2003 and 2006, respectively. From April 2005 to March 2006, he was a Japan Society for the Promotion of Science (JSPS) Research Fellow (DC2) at the Department of Mechanical Engineering and Science, Graduate School of Engineering, Kyoto University. From April 2006 to March 2007, he was a JSPS Research Fellow (PD) at the same department. From April 2007 to March 2008, he was a JSPS Research Fellow at the Department of Systems Science, Graduate School of Informatics, Kyoto University. From April 2008 to March 2010, he was an Assistant Professor at the Department of Human and Computer Intelligence, Ritsumeikan University. From April 2010 to March 2017, he was an Associate Professor at the same department. From September 2015 to September 2016, he is a Visiting Associate Professor at the Department of Electrical and Electronic Engineering, Imperial College London. From April 2017, he has been a Professor at the Department of Information and Engineering, Ritsumeikan University. From April 2017, he has been a visiting general chief scientist, the Technology division of Panasonic, as well. He has been engaged in machine learning, emergent systems, intelligent vehicle, and symbol emergence in robotics.

Lecture: Generative Models for Symbol Emergence based on Real-World Sensory-motor Information and Communication

Music and language have structural similarities. Such structural similarity is often explained via generative processes. This invited lecture introduces the recent development of probabilistic generative models (PGMs) for language learning and symbol emergence in robotics. Symbol emergence in robotics aims to develop a robot that can adapt to the real-world environment, human linguistic communications, and acquire language from sensorimotor information alone (i.e., in an unsupervised manner). To this end, a series of PGMs, including ones for simultaneous phoneme and word discovery, lexical acquisition, object and spatial concept formation, and the emergence of a symbol system, have been developed. This lecture also introduces challenges related to integrating probabilistic generative models and the possible intersection between symbol emergence in robotics and computational music studies.

Suiview: A Web-based Application that Enables Users to Practice Wind Instrument Performance*

Misato Watanabe[1], Yosuke Onoue[1], Aiko Uemura[1], and Tetsuro Kitahara[1]

Nihon University, Tokyo, Japan
chmi19013@g.nihon-u.ac.jp, {onoue.yousuke, uemura.aiko, kitahara.tetsurou}@nihon-u.ac.jp

Abstract. This paper presents a web-based application that enables users to check the stability of the pitches, intensities, and timbres of the sounds they play. Amateur musicians have opportunities to play wind instruments, at a brass-band club at school. To make sounds with the *stable* pitches, intensities, and timbres, players have to carefully control the shapes of their mouth and lips, the strength of the breath, and their vibration. But this is difficult for most amateur musicians, who rely on expert players to check whether they are appropriate and advise them how to improve them. To solve this problem, we have been developing a web-based application to enable amateur musicians to check whether the pitches, intensities, and timbres of their sounds are stable without help from an expert player (https://suiview.vdslab.jp/). In this paper, we describe its basic system design, the current implementation, and preliminary results of its trial use.

Keywords: Wind instrument, Musical practice, Stability, Web application

1 Introduction

Wind instruments are popular among amateur musicians. They are indispensable in brass-band clubs at junior high school and/or high school, and many people enjoy playing a wind instrument as a hobby. However, playing a wind instrument is not easy. To produce sounds with *stable* pitches, intensities, and timbres, players have to carefully control the shapes of their mouth and lips, the strength of the breath, and their vibration.

One problem in learning a wind instrument is a lack of appropriate instructors. In the case of the above-mentioned brass-band clubs at school, the responsible teacher at the club might not be a wind instrument expert. At such clubs, it is often common for novice-level players to teach freshman players. Also, there are fewer music schools that teach wind instruments than the piano.

Wind instrument performances have been investigated from different points of view such as acoustic, psychological, and physiological ones. Brown [1] investigated acoustic features for automatic identification of woodwind instrument sounds. Hirano et al. [3] analyzed muscular activity and related skin movement during French horn performances. Micheal [5] examined the effects of self-listening and self-evaluation in the context of woodwind and/or brass practice by junior high school instrumentalists, and found that self-evaluation was important for improving the instrument.

* This research was supported by JSPS Kakenhi Nos. JP-19K12288 and JP-20K19947.

More recently, there have been attempts to develop systems that allow users to easily understand how their performances are good from visual feedback or computational assessment. Pati et al. [7] applied deep neural networks to automatic assessment of student musical performances. Giraldo et al. [2] developed a system that analyzes sound quality of violin performances and provides visual feedback to users in real time. Knight et al. [4] developed a visual feedback system of musical ensemble focusing on phrase articulation and dynamics. Morishita et al. [6] developed a system that gives novice practitioners (especially children) visual feedback of acoustic features in long-tone training of wind instruments. These systems have been aiming at a goal close to ours, but most of them are not designed to enable anyone to easily check his/her performances on his/her smartphone and/or tablet.

In this paper, we present a web-based application for practicing playing wind instruments by themselves. The important is to give users objective feedback. Because its target users are novice players, we consider that sounds should be stable, in other words, sounds should keep a close pitch, intensity, and timbre from the beginning to the end. Our app. analyzes the pitch, intensity, and timbre of sounds recorded on the app, evaluates their stability, and gives visual feedback to the user. It also provides a function that enables the user's teacher to give comments to the recorded sounds.

2 Basic Design and Functions

Our app aims to provide wind instrument practicers with useful information about the sounds performed by them. For novice-level players, as discussed in the Introduction, acquiring skills for sounding stably is important. Therefore, one of the important functions of our app. is therefore to visualize the stability of the acoustic characteristics (i.e., pitches, intensities, and timbres) of the sounds performed by the user.

Recognizing how well the user is incrementally improving such stability day by day is also important. Therefore, we implement a function for visualizing recording-by-recording variations in the stability of the pitches, intensities, and timbres as well as visualizing the acoustic characteristics of each recording.

Also, we implement a *teacher-to-student comment* function. Although objective visualization is useful for novice players, subjective evaluation and comments by their teacher is also important. By linking a teacher-mode user to student-mode users, the teacher-mode users can listen to the recordings of the linked student-mode users and give them his/her evaluations and comments.

2.1 Recording

Once the user opens and logs into our app., he/she can select what to play from a *long tone*, a *scale*, and an *arpeggio* (Fig. 1). The scores displayed are shown in Fig. 2. After selecting one from these three scores, the user starts recording his/her performance with a sampling rate of 48 kHz (Fig. 3). Recorded sounds are automatically stored on our web server with some metadata such as the user ID, and the recording date.

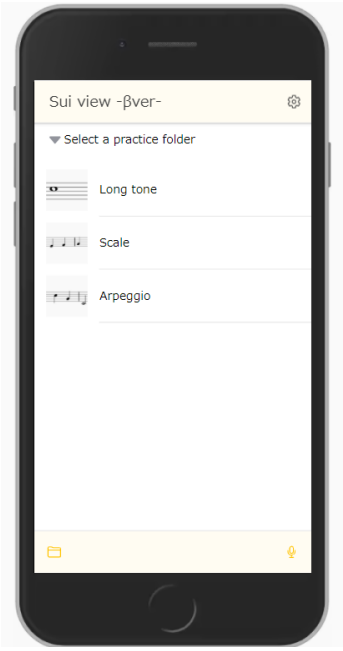
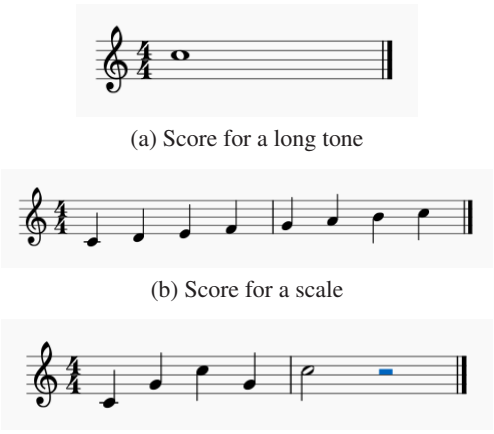


Fig. 1. Screen for selecting what to play



(a) Score for a long tone

(b) Score for a scale

(c) Score for an arpeggio

Fig. 2. Three scores currently supported by our app.

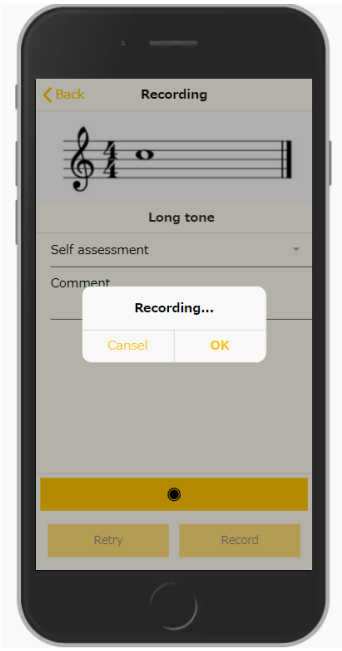


Fig. 3. Screen for recording a sound

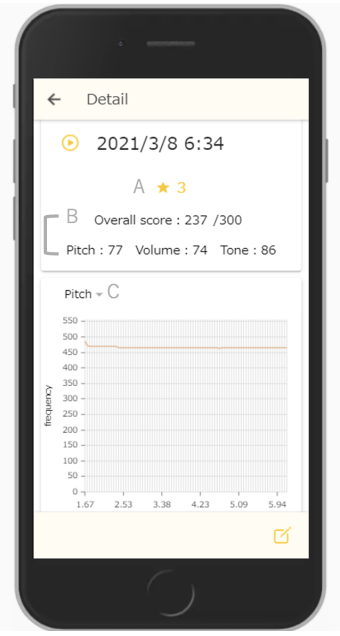
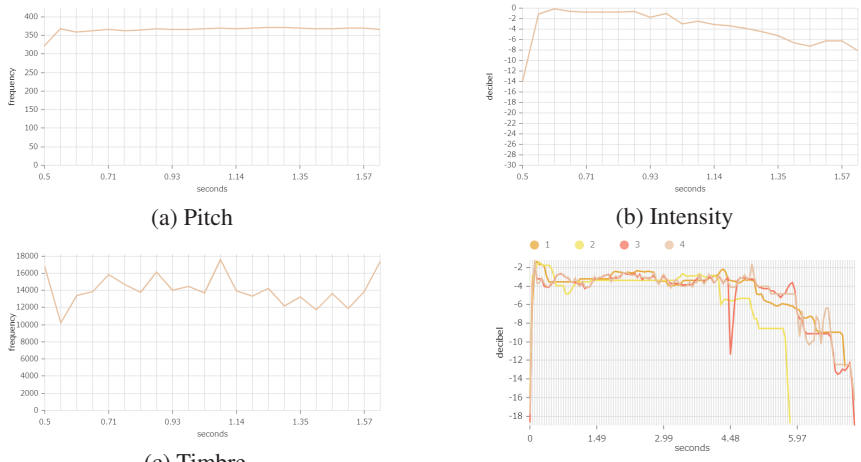


Fig. 4. Example of analysis results (A: self assessment, B: stability scores, C: chart)



(a) Pitch

(b) Intensity

(c) Timbre

(d) Intensities for multiple sounds

Fig. 5. Examples of visualization of acoustic features of recorded sounds

2.2 Visualizing the acoustic characteristics

Once a recording is stored on the webserver, its acoustic analysis starts. The fundamental frequency (F0), amplitude, and spectral roll-off are extracted with a 512-point shift from the recorded sound. We use Librosa (https://librosa.org/) for extracting these features. Next, these features are plotted on the screen, as shown in Fig. 5 (a) to (c). Features for multiple sounds can be plotted on the same screen, as shown in Fig. 5 (d).

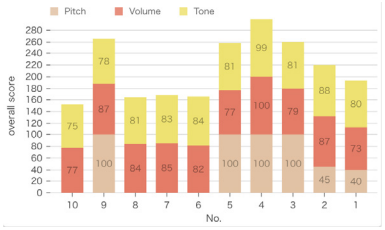
2.3 Visualizing the recording-by-recording variations in the stability

The stability of the pitch (F0), intensity (amplitude), and timbre (spectral roll-off) is calculated for each recording. The stability is defined based on the temporal standard deviation of each feature. Let $\sigma_{\mathrm{F0}}, \sigma_{\mathrm{Amp}}, \sigma_{\mathrm{Sp}}$ represent the temporal standard deviations for the F0, amplitude, and spectral roll-off, respectively. Then, their stability s_i ($i \in \{\mathrm{F0}, \mathrm{Amp}, \mathrm{Sp}\}$) is defined as $s_i = 100 \exp(-\sigma_i / a_i)$, where a_i are pre-defined constants ($a_{\mathrm{F0}} = 4, a_{\mathrm{Amp}} = 70, a_{\mathrm{Sp}} = 1500$). Thus s_i has a value between 0 to 100.

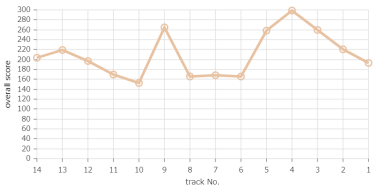
The stability is visualized in two ways to enable the user to check the stability for multiple recordings at a glance (Fig. 6). One is a stacked bar chart that represents the stability of each of the pitch, intensity, and timbre (Fig. 6 (a)). The other is a line chart that represents overall stability scores (Fig. 6 (b)).

2.4 Teacher-to-student comment

Logging in with the teacher mode, the user can listen to sounds recorded by the linked student-mode users and check the visualization of their acoustic features and stability scores. Also, using the teacher-mode, the user can write comments. The comments are automatically sent to the corresponding student-mode user.



(a) Stacked bar chart (for each stability)



(b) Line chart (for total stability score)

Fig. 6. Examples of visualization of Recording-by-recording stability variations

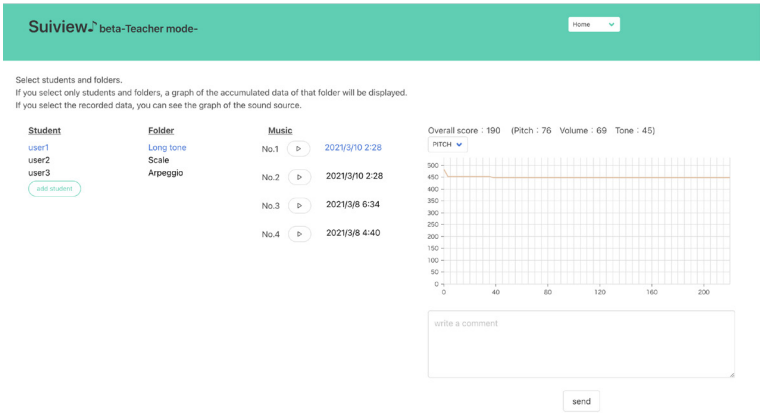


Fig. 7. Screen for the teacher mode

3 Trial Use

Three participants used our app for a preliminary evaluation of the effectiveness of the app. Out of the three participants, one (P 1) was an active player with an intermediate-to-advanced level while the other two (P 2 and P 3) were novices, though they had experience in playing instruments in the past.

Logging in with the student mode, the participants played a long tone, a scale, and an arpeggio on the clarinet several times and recorded them on our app. They saw the visualization of their sounds made by our app. and were asked to answer the following questions on a four-level scale (4: agree, 1: disagree):

Q1 Do you think this app helps you produce stable sounds?
Q2 Did you get useful information from the visualization?
Q3 Are the stability scores close enough to your own impression?

The results, listed in Table 1, imply that the participants comparatively highly evaluated our app. In fact, the two novice-level participants gave us comments such as:

– By listening alone, it was difficult to find what to improve to produce stable sounds.

Table 1. Results of the preliminary questionnaire (1 to 4)

	P 1	P 2	P 3
[Q1] Do you think this app helps you produce stable sounds?	3	4	3
[Q2] Did you get useful information from the visualization?	2	4	4
[Q3] Are the stability scores close enough to your own impression?	2	4	4

- With graphical visualization , novice-level players could find what to improve.
- Line charts were easy to grasp which were good and which were not.

On the other hand, one participant answered that he/she could not understand what each graph means. More intuitive visualization should be explored. We also received an opinion that they wanted to see the analysis for sounds given by professional players.

4 Conclusion

In this paper, we presented a web-based application that enables users to recognize the stability of wind instrument sounds played by them by visualizing their acoustic features and stability scores. Once the user records his/her wind instrument sounds on the app, their acoustic features including the pitches, intensities, and timbres are analyzed as well as their stability is evaluated. Three participants in a preliminary experiment gave us comments that the visualization was useful to produce stable sounds.

Although we focused on the stability of pitches, intensities, and timbres, more complex expressions such as detailed dynamics would be important for more advanced players. We will extend the app to support such advanced level players' practice as well as systematic evaluation of our app.

References

1. Brown, J.C., Houix, O., McAdams, S.: Feature dependence in the automatic identification of musical woodwind instruments. The Journal of the Acoustical Society of America **109**(3), 1064–1072 (2001)
2. Giraldo, S., Ramirez, R., Waddell, G., Williamon, A.: A real-time feedback learning tool to visualize sound quality in violin performances. In: Proc. of MML 2017. pp. 19–24 (2017)
3. Hirano, T., Kudo, K., Ohtsuki, T., Kinoshita, H.: Orofacial muscular activity and related skin movement during the preparatory and sustained phases of tone production on the french horn. Motor Control **17**(3), 256–272 (2013)
4. Knight, T., Bouillot, N., Cooperstock, J.R.: Visualization feedback for musical ensemble practice: A case study on phrase articulation and dynamics. In: Proc. VDA 2012 (2012)
5. Michael, P.H.: The effects of modeling, self-evaluation, and self-listening on junior high instrumentalists' music performance and practice attitude. SAGE journals **49**(4), 307–322 (2001)
6. Morishita, T., Oguchi, H., Kunimune, H., Kirihara, A., Honma, Y.: Development of a support system for beginners to practice long-tones in school wind instruments. Kyoiku Jissen Kenkyu, Shunshu University Higher Education System Center (2018), (in Japanase)
7. Pati, K.A., Gururani, S., Lerch, A.: Assessment of student music performances using deep neural networks. Applied Sciences **8**(4) (2018)

Continuous Parameter Control Using an On/Off Sensor in the Augmented Handheld Triangle

Marcio Albano H. Ferreira, Tiago F. Tavares

School of Electric and Computer Engineering - University of Campinas (UNICAMP) - Brazil
marcio.ahf@gmail.com, tiagoft@gmail.com

Abstract. We present Triaume, a handheld augmented non-pitched percussive musical instrument based on the triangle. Our proposal relies on a capacitive thumb sensor, which allows controlling digital musical devices while preserving the possibility of playing the instrument's traditional techniques. We reduce the augmentation invasiveness by using an external smartphone to emulate faders related to the instrument's configurations. Triaume's interaction proposals are built from idiomatic techniques used in regional Brazilian music genres. We can use the sensor as an on/off button that can, either on touch or release, trigger pre-programmed percussive sounds that can be played together with the triangle's acoustic sound. Also, we use a low-pass filter to convert the digital sensor's acquisitions to a continuous value, allowing expressive synthesis control. Triaume can be used in avant-garde music, and its interaction design favors its use in variations of traditional music.

Keywords: Triangle, Capacitive sensor,Pulse Width Modulation (PWM),Augmented instrument, Brazilian music

1 Introduction

Traditional music instruments can be augmented with electronic sensors, which can acquire signals to control devices like synthesizers and effect processors. These sensors usually exploit the so-called spare bandwidth [1] , that is, movements or limbs that are not used in the traditional playing techniques and, therefore, can be used for other purposes. Augmented instruments can provide new expressive possibilities when compared to their traditional counterparts.

This work presents an augmentation proposal for the triangle, a handheld non-pitched percussion instrument traditionally used in several regional Brazilian music genres such as Forró, Xote, and Baião [2] . The acoustic triangle is usually held with one hand using the index finger and played with the other hand using a metal mallet. The instrument's sound can be damped by closing the holding hand's palm around the triangle's side.

Our augmentation proposal uses a single capacitive sensor [3], [4], [5], [6], [7], [8] placed on the instrument's upper corner. The sensor is isolated from the instrument's body and is activated with the holding hand's thumb independently of the damping or mallet striking actions. This placement allows an interplay between the traditional techniques and the augmented possibilities.

This minimalistic augmentation barely impacts the use of the traditional techniques but brings other challenges of its own. The first one is to allow the musician to configure the instrument's parameters during performance without bringing a computer to the stage. We mitigate this problem using a smartphone, which provides all necessary faders for this configuration. The second challenge is to provide a diversity of interactions that can be creatively explored.

To tackle this challenge, we use mapping strategies inspired in the common (even if not traditional) technique of playing other instruments, such as hi-hats (triggered with a pedal), or sets of cowbells or carillons [9], [10], together with the triangle in Brazilian regional music. In our proposal, we investigate the possibilities of triggering events on sensor touch or on sensor release, inspired by the damp (close hand) and release (open hand) gestures typically used in Forró music. They can be used to play virtual instruments, in special percussive sounds, allowing the musician to play with more instrumental layers.

Additional control possibilities can arise from encoding the sensor information through time [11] . In our proposal, we use a low-pass filtering technique to convert a sequence of on/off acquisitions to a continuous control signal, similarly to a Pulse Width Modulation motor control [12], [13] . This allows using the capacitive sensor as an interactive fader that can be controlled using rhythm.

2 Instrument Design

The triangle augmentation consists of three blocks, as shown in Figure 1. The first is Triaume itself, which is a regular acoustic triangle with an attached capacitive sensor and an ESP32 microcontroller [14]. The second is a smartphone that runs a MobMuPlat [15] patch and controls the digital configurations. Both of these blocks send Open Sound Control (OSC) [16] packets to the third one, a computer that executes sound synthesis and control in a Pure data (Pd) patch [17]. Each of these blocks is discussed next.

2.1 Triaume Body

The augmented triangle has one single sensor, which is a capacitive sensor attached to the triangle's upper corner. As shown in Figure 2 , the sensor is isolated from the instrument's body using insulating tape. A distance was kept between the insulated tape covered area and the region that is normally stroke by the triangle mallet when applying techniques used in the context of Brazilian music. Mounting the sensor close to the triangle's tip reduces the sensor's impact on the sound's quality.

We used the Capacitive Sensor library created by Paul Badger [18], [19], which allows to build high sensitivity sensors using only a resistor, a microcontroller, and an electrode, which can be made of any conductive material. Our electrode was made using copper tape and it was connected to a 1MΩ resistor, linked to one of the ESP32 pins. The library continually yields capacitance readings, which are disturbed by touching the copper tape.

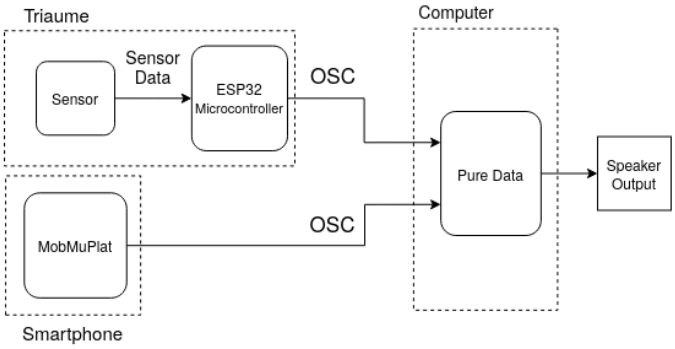


Fig. 1. System overview. The computer receives OSC packets both from the Triaume and from a smartphone.

Both the sensor type and microcontroller model are chosen based on the idea of developing a low-cost instrument since the acquisition of imported products in Brazil is expensive due to taxes. In some cases, the final cost of imported products can reach even twice the value of the original cost. [20]. Therefore, using low-cost components is desirable for allowing easy access to the instrument.

The microcontroller is attached to the musician's body, reducing its impact on the instrument's sound and playability. It sends the measured capacitance values to the computer using OSC packets, which can use a RS-232 connection with serial line internet protocol (SLIP) [21] or UDP packages over a Wi-Fi connection. The RS-232 connection provides a lower delay, but requires a connection cable; conversely, the Wi-Fi connection allows a greater mobility for the musician, but tends to have longer and more unstable delays [22]. This simple setup is barely invasive to the instrument but requires an additional device to provide configuration faders for performance usage, as described next.

2.2 Smartphone

It is often desirable that control-to-sound mapping proposals allow on-site adjustments, either during soundcheck or to change sonorities in different parts of a performance. Our system provides this functionality using a smartphone application, shown in Figure 3. The application is based on MobMuPlat and sends the computer configuration parameters using OSC over WiFi. Similarly to the knobs in a guitar effects pedal, the application can be used intermittently on stage.

The advantage of using a software application is that it can be easily configured and expanded as to match different sound processing proposals that might be built using Pd. Moreover, because it is external to the triangle, it can be left in a safe place during performance. Henceforth, this design option contributes to reduce the invasiveness and flexibility of Triaume's setup when compared to the idea of having physical knobs attached to the triangle.

Fig. 2. The capacitive sensor is attached to the triangle's tip and isolated from the instrument's body using insulating tape.

2.3 Computer

The final block in the augmentation system is a computer, which executes a Pd patch responsible both for converting the capacitive sensor's continuous values to on/off information and for synthesizing audio to be played in a loudspeaker.

It is important to note that the conversion from continuous to on/off values could be performed in the microcontroller. However, this conversion depends on a threshold that changes depending on the sensor's materials, electric noise, the instrument's shape, and the size of the musician's hands. Therefore, this conversion is performed in the computer, and the threshold is configured using the smartphone, as described in the previous section.

The on/off sensor information is used to control sound synthesis using two different strategies, as shown in Figure 4. For the first strategy, the sensor touch and release gestures are immediately mapped into on/off information for sound activation. In the second one, the on/off information is low-pass filtered, thus providing a continuous sound parameter control. Each of these strategies is discussed next.

Using On/Off Information for Sound Activation A simple, immediate control strategy is to map the on/off sensor to a synthesizer's ADSR envelope controller. This allows using the sensors as a key that triggers and sustains a particular sound. The sensor (and, consequently, the related sound) can be played independently of the triangle's damping because it uses the thumb while the damping process uses the hand palm.

Although the sensor can provide the musician with another sound layer, it can be hard to physically combine it with muting the triangle with the hand palm. For some rhythmic patterns, it can be easier to play sounds when the sensor is released. It is possible to reach a myriad of rhythmic possibilities by combining the different activations (on touch/on release) with sound synthesis configurations.

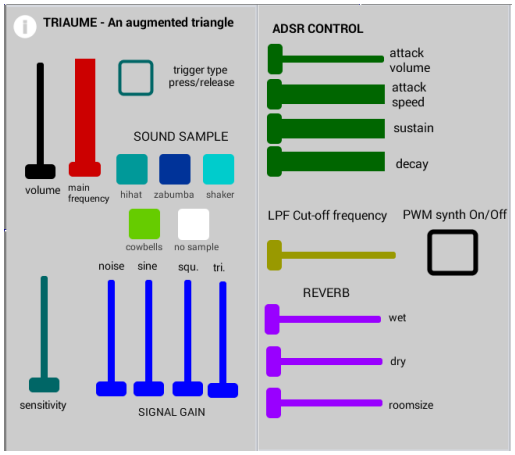


Fig. 3. Smartphone app graphical user interface

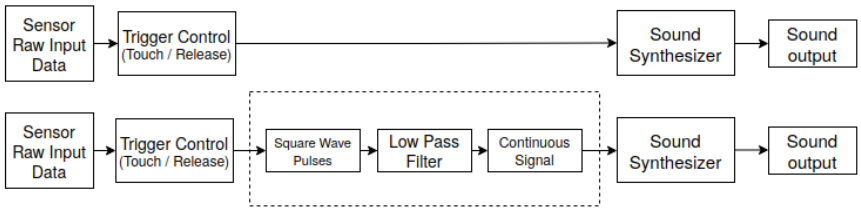


Fig. 4. Mapping strategies. The on/off sensor can be used either on touch or on release (left). Also, a low-pass filtering technique can convert the on/off information to a continuous signal (right).

Figures 5 and 6, respectively, illustrate both of these activation modes, showing the on/off sensor data (upper panel) and the corresponding synthesized waveform. In both cases, the parameters for attack speed and sustain were adjusted for minimal values, which highlights the synchrony between the input signal and the sound output.Next, we present our proposal to generate continuous control with the sensor.

Continuous Parameter Control: a PWM-like Approach Continuous controls can be used in expressive sound control in important parameters that can not only be driven directly by a binary event logic, like wet/dry levels, gains, and filter cut-off frequencies. These parameters are usually controlled using faders, knobs, or sensors such as accelerometers. In this section, we describe how to use the on/off sensor to provide continuous control values.

The technique employed obtains continuous values from digital inputs using low-pass filtering, similarly to using Pulse Width Modulation (PWM) [12], [13] control. In PWM, the input signal is a square wave, which is filtered so that the output signal

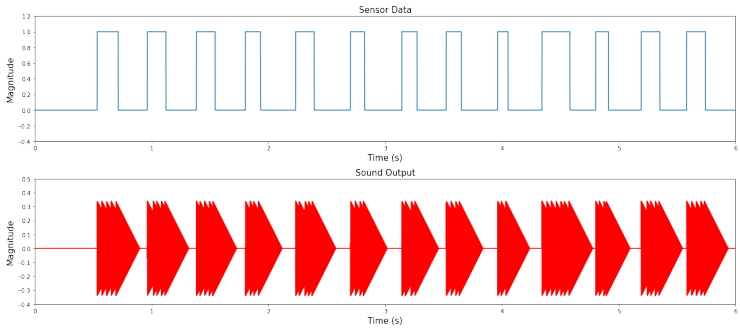


Fig. 5. Sensor data and sound output using trigger on touch.

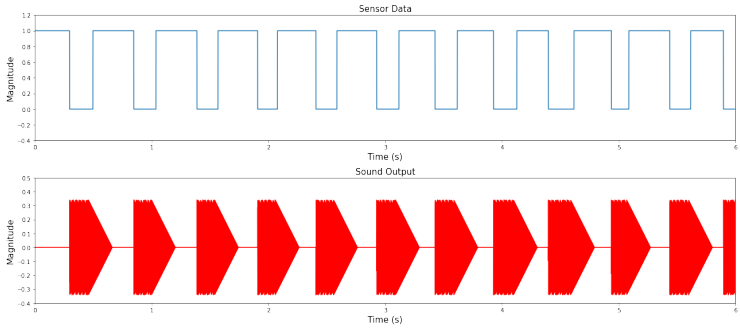


Fig. 6. Sensor data and sound output using trigger on release.

level is proportional to the fraction of time in which the input is high (that is, the duty cycle). Equation 1 shows the relationship between the output signal level (Vout), the value corresponding to high level signal (A), and the duty cycle (d).

$$V_{out} = A \times d \tag{1}$$

In the on/off sensor, we can generate duty cycle variation by intermittently touching and releasing the capacitive sensor. Low-pass filtering generates a smooth, continuous signal, whose level is proportional to the duty cycle. Lower filter cut-off frequencies lead to smoother signals but also to slower responses.

Figure 7 illustrates a demonstration of this technique. It shows an acquisition of the on/off signal and the corresponding output after using a low-pass filtering with cut-off frequency of 0.1 Hz. It can be seen that the filtered output increases accordingly to the duty-cycle and can generate intermediate values wth some ripple.

This technique allows controlling effect or synthesizer parameters using a rhythmic input generated by touching and releasing the sensor. This is especially desirable because it allows using gestures that are close to those native to the Forró music repertoire, that is, playing rhythms with the hand. Moreover, touching and releasing parts of

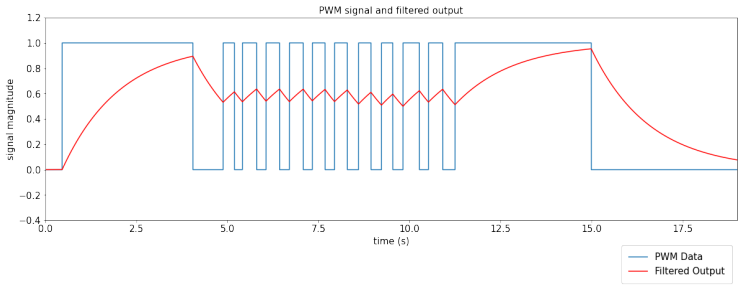


Fig. 7. Data acquired while playing the augmented instrument and low-pass filtered output.

the instrument is also part of the traditional repertoire of many percussion instruments; hence this technique can be applied in other types of drums and other music genres.

Interestingly, both mapping techniques can be combined, generating a sound trigger that is simultaneous to a timbre control. This one-to-many mapping can generate new expression possibilities that do not necessarily fit the regional music genres Triaume was inspired in. The next sections will present tests made for instrument evaluation, followed by comments regarding the possibilities obtained by its use.

3 Instrument Evaluation

We qualitatively evaluated our instrument aiming to identify some of its musical possibilities. Triaume was evaluated from the author's viewpoint, using their own musical experience, first focusing on the on/off sensor, then on the PWM-like control.

3.1 On/Off Sensor

As a first experiment, we programmed Triaume synthesizer to play a sample of a percussive sound triggered by sensor release. A short track was recorded, and a part of its waveform can be seen in Figure 8. The higher magnitude pulses correspond to the synthesized sound and the lower magnitude ones to the acoustic triangle.

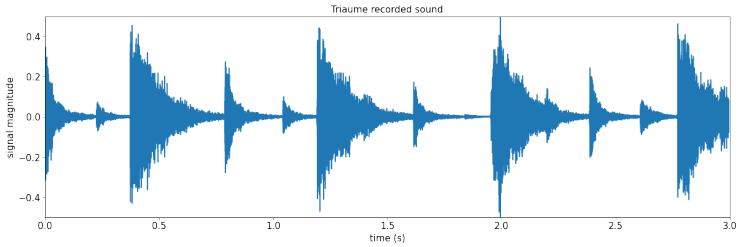


Fig. 8. Triaume sound record triggering a synthesized sound sample..

In this test, the on/off trigger provided a quick response, which allowed playing rhythms without a perceptible delay. The capacitive sensors have shown a high sensitivity and were able to detect even more subtle touches. At the same time, the sensitivity control mechanism allowed rejecting false positives in this detection.

In order to illustrate another musical possibilities, an audio demo was recorded choosing different percussion sounds, and also synthesized sine, square and triangle waves. An interesting outcome was obtained when using an alternate toggle mechanism to play two different cowbell sounds[1] [2].

3.2 PWM-like Control

The PWM-like mechanism test consisted of linking the continuous control mechanism to a FM synthesis control module implemented in Pd. Continuous values control the modulating wave magnitude in the FM synthesis. Adjusting the the low-pass filter cut-off frequency allows tuning continuous signal's change rate speed.

Mapping these continuous values directly into synthesizer fundamental frequencies would lead to an obvious one-to-one mapping strategy [23]. Using such a strategy can lead to results next to the ones obtained when playing the Theremin, but with an inevitable ripple (as suggested by Figure 7).

For demonstration purposes, a song was composed and recorded by the authors in order to show the new instrument application context. This song, entitled "Forró do OSC", shows that the instrument can be used either inside the forró idiomatic, or for avant-garde music[3][4].

4 Discussion

The results presented in this work demonstrate that Triaume can potentially bring new expressive possibilities to the triangle. Its interactions were designed aiming at a low invasiveness regarding the instrument's traditional techniques. Even though one of the authors plays Forró percussion, we could not perform any evaluation with external musicians due to the ongoing COVID19 crisis. However, the song composed by the authors shows an idiomatic Forró example, and at the same time, innovative possibilities for other music genres.

The idea of using low-pass filtering to convert on/off signals to continuous values is not novel per se, as it is a straight implementation of classic PWM control [24]. However, our proposal generates the input signal from a touch sensor placed so that it

[1] Audio demo with percussive sound samples available at: https://soundcloud.com/marcio-albano/triaume-test-samples/s-k7MKiggI4E2

[2] Audio demo with synthesized waves available at: https://soundcloud.com/marcio-albano/triaume-demo-sine-triangle-square-waves/s-kzXDBiI6izr

[3] Song available for listening at https://soundcloud.com/marcio-albano/forrodoosc/s-Qty5RREnsph

[4] "Forró do OSC" music score available at https://1drv.ms/b/s!AnEYggKX1_PYkq4n8-k8xKsa44XXBA?e=Wc6mPG

captures rhythms in the context of a handheld percussion. Henceforth, this process can be interpreted as a rhythm-to-control conversion, which can be applied in several other instruments.

5 Conclusion

This work presents an augmentation for the triangle, a handheld non-pitched percussion instrument used in several regional Brazilian music genres. The augmentation proposal uses minimalistic and low-cost hardware, comprised of a single capacitive touch sensor attached to the triangle's upper tip, which reduces its impact on using the traditional techniques to play the triangle. On-stage configuration possibilities are obtained by using an external mobile device to fine-tune all parameters.

We use two mapping strategies. The first one uses the capacitive sensor as a key, which can operate either on touch or on release, making it possible to add another instrumental layer to the acoustic one. The second one uses a low-pass filtering technique to convert the on/off information to a continuous control, which allows reconfiguring synthesis or digital effect parameters by performing rhythms with the thumb.

Although the on/off to continuous signal conversion was inspired by the use of the triangle inside the Forró music context, it can be used in other musical instruments and genres, e.g., using the gestures related to touching a drum's membrane or side to change its resonance. The main idea of the sensor is to convert rhythmic interactions to a continuous value, that is, it uses gestures that are native to playing percussions. Hence, the proposed augmentation is not only useful for Forró music itself, but also a potential path to augment other percussive instruments in other genres.

In future work, in addition to the "Forró do OSC" song composed for this work demonstration, we will seek to present the augmented instrument to contemporary music bands so that it can be explored and further improved. Currently, this process is strongly harmed by the COVID-19 crisis, which brings forward the problem of developing musical hardware in collaboration with musicians without physical social contact.

Moreover, further sound exploration can be made using the ripple present on the low-pass filtered output signal. This approach could give the instrument more expressiveness when used with adequate mapping strategies.

References

1. Perry R. Cook. Principles for designing computer music controllers. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 3–6, Seattle, WA, 2001.
2. Dominique Dreyfus. *Vida do viajante: A saga de luiz gonzaga.* Editora 34, 2012.
3. Max V. Mathews. The radio baton and conductor program, or: Pitch, the most important and least expressive part of music. *Computer Music Journal*, 15(4):37–46, 1991.
4. Colin Honigman, Jordan Hochenbaum, and Ajay Kapur. Techniques in swept frequency capacitive sensing: An open source approach. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 74–77, London, United Kingdom, June 2014. Goldsmiths, University of London.

5. Enric Guaus, Tan Ozaslan, Eric Palacios, and Josep L. Arcos. A left hand gesture caption system for guitar based on capacitive sensors. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 238–243, Sydney, Australia, 2010.
6. David Gerhard and Brett Park. Instant instrument anywhere: A self-contained capacitive synthesizer. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.
7. Nan-Wei Gong, Nan Zhao, and Joseph Paradiso. A customizable sensate surface for music control. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.
8. Diana Young and Georg Essl. Hyperpuja: A tibetan singing bowl controller. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 9–14, Montreal, Canada, 2003.
9. Banda Falamansa. Sete meninas / forró do bole bole - mtv ao vivo. `https://www.youtube.com/watch?v=BzwqMciFc-s`. [Online; accessed on 14-June-2021].
10. Baião de Rua. Baião de rua no release showlivre. `https://www.youtube.com/watch?v=JLg-c0ov7sI`. [Online; accessed on 14-June-2021].
11. Gabriel Lopes Rocha, João Teixera Araújo, and Flávio Luiz Schiavoni. Ha dou ken music: Different mappings to play music with joysticks. In Marcelo Queiroz and Anna Xambó Sedó, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 77–78, Porto Alegre, Brazil, June 2019. UFRGS.
12. Daniele Lacamera. *Embedded Systems Architecture*. Packt Publishing, 2018.
13. Jonathan W. Valvano. *Embedded Systems: Real-Time Interfacing to Arm® Cortex™-M Microcontrollers*. 2014.
14. Espressif esp32. `https://www.espressif.com/en/products/devkits`.
15. D. Iglesia. The mobility is the message : the development and uses of mobmuplat. 2016.
16. Adrian Freed and Andrew Schmeder. Features and future of open sound control version 1.1 for nime. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 116–120, Pittsburgh, PA, United States, 2009.
17. Miller Puckette. Pure data. https://puredata.info/. [Online; accessed on 14-June-2021].
18. Paul Stoffregen. Arduino capacitivesensor library (repository). `https://github.com/PaulStoffregen/CapacitiveSensor`. [Online; accessed on 14-June-2021].
19. Paul Badger. Arduino capacitivesensing library. `https://playground.arduino.cc/Main/CapacitiveSensor/`. [Online; accessed on 14-June-2021].
20. Romulo A Vieira and Flávio Luiz Schiavoni. Fliperama: An affordable arduino based midi controller. In Romain Michon and Franziska Schroeder, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 375–379, Birmingham, UK, July 2020. Birmingham City University.
21. CNMAT. Osc: Arduino and teensy implementation of osc encoding. `https://github.com/CNMAT/OSC`. [Online; accessed on 14-June-2021].
22. Geise Santos, Johnty Wang, Carolina Brum, Marcelo M. Wanderley, Tiago Tavares, and Anderson Rocha. Comparative latency analysis of optical and intertial motion capture systems for gestural analysis and musical performance. [accepted for publication at NIME 2021].
23. Andy D. Hunt, Marcelo M. Wanderley, and Matthew Paradis. The importance of parameter mapping in electronic instrument design. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 88–93, Dublin, Ireland, 2002.
24. Bishop R. H. Dorf, R. C. *Modern control systems*. Upper Saddle River, NJ, Prentice Hall., 2014.

Locus Diffuse: An Agent-Based Sonic Ecosystem for Collaborative Musical Play

Rory Hoy and Doug Van Nort

DisPerSion Lab rorydavidhoy@gmail.com, vannort@yorku.ca

Abstract. Locus Diffuse is a networked multi-user instrument populated by a simulated slime mold and four human players. Mimicking the biological behavior of slime mold and establishing a virtual living network between player nodes, the system sonifies interaction along these connections. Participants use a browser based interface to play the multi-user instrument, and access an accompanying stream for audio and visual output of the system. Player responses from various play sessions are explored and reported in relation to sonic ecosystems as a product of sound sources intersected with agent behavior, defining interaction through personal connection to agents, an aural vs visual understanding of the system, and various frames of focus employed by participants in regard to human/machine and inter-human collaboration.

Keywords: agent-based musical systems, multi-user instruments, natural computing, slime mold

1 Introduction

Musical play has acted as a vessel for a communal engagement, identity, exploration, and expression throughout history [6]. While the style of play may vary from recital of composed works to free improvisation (and every permutation in between/beyond), a common thread is that emergent group playing dynamics are revealed through the complex interactions between each player [2]. This aspect of musical collaboration is a social ritual in which participants are afforded a medium of aural communication beyond the verbal. Players can be represented as nodes within a network of participants that expresses interpersonal playing decisions, and the resulting sonic landscape can be seen as an emergent form of this established network. Viewed in this way, collective action results in a cumulative sound field that is the product of each node's (player's) input. An interactive instrument/environment, named *Locus Diffuse* was developed to investigate and facilitate these emergent participatory network structures within collaborative musical play for four players. This is mediated by an instrument in which users can "play" a space through interaction both with its population of simulated agents and with each other. Situated at the crossroads of sonic ecosystem design, agent-based musical systems, multi-user instruments, and networked performance, *Locus Diffuse* draws on a network of practices to produce a system that is used to interrogate the outcome of their resulting collaborative human/machine interplay. The system was initially planned for a full scale room implementation within the DisPerSion Lab at York University, however due to social distancing restrictions caused by the global COVID-19

pandemic, the project was required to pivot to a distributed virtual performance space. Players and spectators access a live audio/visual stream as a collective hub for generated activity, while controlling their input within an additional browser window or mobile device. During this time of relative isolation, the project's aesthetic themes of connection and collaboration were heightened through this additional networking component, facilitating the communal play of all participants.

The behavior of the system's population of agents is modelled on networking structures found within the biological form of slime mold. Harnessing natural processes of emergent form and community, these organisms have been demonstrated to have repeatable emergent behaviors of aversion and attraction to environmental stimuli. Most notably their structure takes the form of thin physical networks between food sources, and through implementing approximations of this behavior, *Locus Diffuse* generates flowing and reactive networks of autonomous agents moving between player positions. We argue that these organisms are well suited as a metaphorical frame that mirrors the collaborative generative network-like structure found within musical performance, and that mapping various interaction responses can result in compelling ecosystemic behavior.

2 Related Works & Literature Review

2.1 Harnessing Biology - Artistic & Computational Implementations

Natural Computing studies the application of natural phenomena within ecological systems and biological structure to a multitude of computational tasks [18]. These implementations can come in the form of mimicry, approximation, and inspiration from structures found within natural systems. Slime mold, specifically Physarum polycephalum, exhibits extraordinary behavior for an organism which contains no explicit sensory organs, capable of tactile, chemical, and photoreceptive sensing. The body consists of a single cell, but can produce many flexible space-searching tubules and can change their thickness to allow for a greater flow of cytoplasm in order to move. The body attempts to move in a direction towards food/positive stimulus or away from negative stimulus [5]. The slime mold is able to then retract, reinforcing a minimal path between all available food sources within even complex spatial layouts such as mazes [14]. Computational models of slime mold have resulted in creating logical gates, solving resource heavy computation, and achieving primitive memory [1]. Artistic applications of slime mold have been advancing in tandem with computational implementations. Miranda et al. [12] constructed a sound synthesis project which allowed for recordings of voltage at various locations through the electrical activity of a slime mold network across a series of food nodes. This data was then used within a granular synthesis engine to generate sonic events.

2.2 Sonic & Performance Ecosystems

Sonic ecosystems refer to interactive systems defined by the generation of a reactive audio environment in which self observing behavior and participant input result in audible

dynamic feedback [4]. Such systems explore the relationships and outcomes established between human, machine, and ambient environment. A central question in the context of ecosystemic design is the role of the human participation within an established work, and what constitutes "interaction". Some systems generate a sonic environment purely mediated by an established machine/ambience relationship, while others find room for human interaction to extend these interactions. Di Scipio [4] describes this ability of system self observation as "a shift from creating wanted sounds via interactive means, towards creating wanted interactions having audible traces", and claims that it is through these traces that compelling sonification can occur.

The original, in-person formulation of *Locus Diffuse* was initially planned to play off of the self-observing vocal & ambient feedback found within the design of the *dispersion.eLabOrate* project [9], a system exploring collaborative sounding within a Deep Listening-inspired sonic meditation [15] context. Within *Locus Diffuse*, self-observation occurs at the agent level. Each agent is only aware of its own state (vs a sense of other's or environmental current states) and acts according to its sensory input from the environment. Environmental changes and subsequent sonification are a result of the interplay between players and the system's agents.

2.3 Multi-User Instruments & Networked Music

Intended to promote close relationships between multiple players and resulting play techniques, multi-user instruments allow many participants to perform through a singular instrument. Designing for a multi-user instrument context requires explicit consideration of the intricacies and collaborative experiential content which the instrument/system needs to convey. Jordà [11] outlines key aspects of multi-user instruments that facilitate shared collective control within a musical system. These properties include number of users, user roles, player interdependencies/hierarchies, and the flexibility of each of these components.

Creation of mutual-influence via networked sound data has been explored by pioneering groups such as the League of Automated Music Composers and The Hub [8]. More recently, these networks have also been explored within the realm of telematics, employing the internet as a medium for musical collaboration [16]. Weinberg [21] presents the concept of an Interconnected Musical Network (IMN), live performance collectives in which player interdependencies result in dynamic social relationships and reactive playing. Weinberg states a successful musician network would promote "interpersonal connections by encouraging participants to respond and react to these evolving musical behaviors in a social manner of mutual influence and response", positioning the performance of group-based music as a social ritual. Additionally, exploring a biological metaphor of the established network, Weinberg [21] states: "Such a process-driven environment, which responds to input from individuals in a reciprocal loop, can be likened to a musical 'ecosystem.' In this metaphor, the network serves as a habitat that supports its inhabitants (players) through a topology of interconnections and mutual responses which can, when successful, lead to new breeds of musical life forms...". This parallels the key ecosystemic theme of *Locus Diffuse* and points back towards the culmination and amalgam of these disparate practices as viable in fostering a connected musical collaborative space.

3 Artistic Intention & System Overview

Locus Diffuse introduces a simulated being that reacts to the movement of players, permeating the environment as a traversable medium. Sonification of the system is achieved when interacting with this mediating entity as well as through participant movement in virtual space, and can therefore only exist/function through the symbiotic relationship of players to it and to each other. Control is not centralized to one participant, nor surrendered to the simulated organism. This control facilitates the musical composition of space, sculpting a form which the simulated organism populates spatially and aurally. This emergent structure and reactive behavior can be paralleled within the participants of the social ritual of "musicking" [19], in which each player has a sensory experience of the whole while also contributing to it. Participating within the shared audio space means enacting this social ritual of musical play, thus the roles and capabilities of players along with the function of environmental agents were established such as to rely on all players. These inter-human and human-agent relationships are critical to explore the resulting network structure. The simulation is contained within Max, employing JS for directing agent positions and control data for grain sonification, JWeb in Max is used for visual feedback in an HTML page, and audio synthesis control patches additionally developed within Max.

3.1 Simulated Agents

Agent behaviour is modeled after the biological structures of Physarum polycephalum, but does not represent an exact scientific model of the organism. Player positions are represented as purple radial gradients within the simulation. Player positions act as food deposits for the simulated agents, and movement results in variations of the environmental structure sensed by the collective simulated slime mold. The simulation is informed by the research of Vogel et al. [20] and inspired by Jones [10], who outlines the mechanics of Physarum polycephalum.

An initial population of 500 agents spawn in the centre of the simulated environment and are given a random starting vector. Each agent is equipped with two sensors positioned at an angular offset of 45 degrees left and right, and a set distance ahead of the agent. The simulated world is quite large (1000^2 pixels) in relation to the size of the cellular bodies (2 pixels), necessitating sensors that have a far reach (default 350 pixels), allowing them to "smell" food sources and trails from a reliable distance. As mentioned in Jones [10], this large distance would normally be considered remote sensing separate from the body of an agent, however this distance also acts as the "overlapping actin-myosin mesh of the plasmodium gel system", allowing the cells to understand their position relative to each other and to nutrient sources. Optimization of the agent network is achieved through a decaying chemoattractant trail deposited and sensed by each agent. Trails are deposited when an agent senses food or another trail, resulting in deposits towards food. As trails diminish over time, an established network is strengthened when searching agents return from an unsuccessful search, or travel along the stream, continually depositing additional trails. Agent sensors check for light values representing chemoattractant strength, average the data collected, and then determine the direction to face. Agents remember the last strongest "smell" they've sampled and

choose what to do based on the current reading, always orienting towards the highest value. Agents are in search of energy to keep moving and find more food. Each agent mimics the cytoplasmic streaming behavior of a slime mold, and represents a theoretical main concentration node of this cytoplasm. Energy is a value held by each agent and player attractant node, which maps to qualities of each granular sonification, movement, and rotation speed. Losing energy will cause them to slow or enter a hibernation-like state when approaching zero. Agents which gain energy again can be "revived" from this hibernation state if passed over by a player. Simulated agents actively gain energy while upon a player, while passively losing energy during movement/wandering between nodes. Players regain energy by being in close proximity to others. Agents keep individual energy values as opposed to distributing energy, allowing for unique sonifications based on the amount of energy one contains.

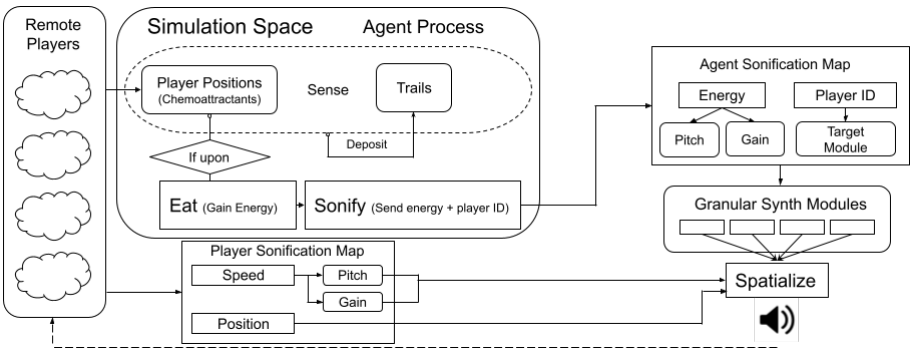


Fig. 1. Human and machine interaction with resulting data flow to sonification

3.2 Sonification

Unique source material was used to ensure an identifiable timbre for each player. Play sessions were done in two waves and audio sources were edited between waves for both refinement of sonification aesthetics, and to gauge changes in play due to these varied timbres. Wave 1 sources were textural in nature, using viscous drips, synth drones, running water, and a filtered conversation as audio material. Wave 2 sources were chosen to result in crisp sonification - timbrally in line with clicking, dripping, droning, and swarming noises. Sonification of a given audio grain was triggered by an instance of an agent "eating" at a particular player location, when an agent takes energy from the player's representational chemoattractant. The sounding potential of a grain triggering is randomized to a 1 in 500 chance upon an agent eating to avoid continuous audio output from a single agent, while also mimicking variance in time needed to break down and process energy from food sources (i.e. a second artistic liberty taken with the model). Messages are sent from the logic JS running in a JWeb, routed to one of four granular synthesis engines, corresponding to a different player. These include the energy value of the agent, and the player ID acting as their source of energy. The granular

synthesis patches contain a Petra buffercloud object [13], allowing for accurate single-grain firing (5-50ms long). Energy values are mapped to a pitch multiplier of the source material and gain level. As energy values range from 0 - 100, values are scaled to an appropriate pitch range multiplier between 0.5(±0.2) and 1.7(±0.2), and gain ranging from -30dB to 0dB. Granular synthesis output is then spatialized to the corresponding player position. Player movement is sonified by high frequency sine tones. Unique frequencies are assigned per player, then modulated based on movement speed, with slower movements being modulated down (with higher gain), and faster movements modulated up (with lower gain), which may produce a beating depending on relational position/speed of multiple players. These tones are spatialized in a virtual binaural space using IRCAM's Spat [3].

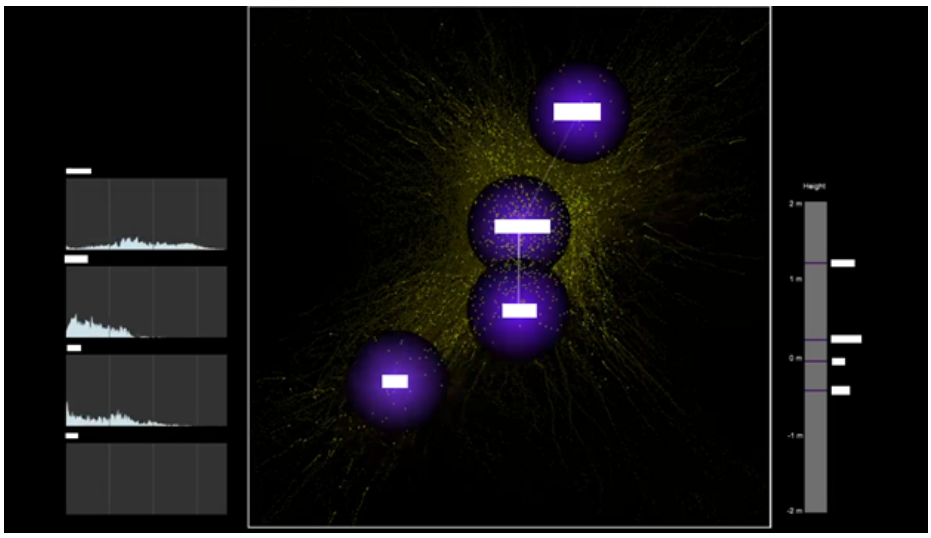


Fig. 2. Stream view of the simulated environment depicting State 4 (participant names censored)

3.3 Networked Interaction & Visualization

Accommodating different devices and network connections was essential in order for public accessibility to players and audience. Control of player movement occurs in the browser through a provided URL, and can be accessed with a browser or touch enabled device. The interface contains a panel for each player consisting of identical controls, including a centre square for position input, and a right-hand slider for vertical movement. The left-hand boxes show spectroscope representations of current sonic activity for each participant. Visual output of the system (Fig. 2) was hosted on a public live stream. This situates the stream as a centralized audio and visual hub for the experience of the instrument, and resulting sonic ecosystem.

4 Survey

Play sessions were held in two waves as open calls on set dates, and public exhibitions following the weekly electro-acoustic improvisation series *DisPerSion Relation X*. The first wave of play sessions focused on a single behavior state of the agents, while the second wave presented players with four varied states. Sessions lasted roughly 25 minutes (with some lasting up to 60 minutes). Following play, participants were asked to complete an anonymous web form. As Wave 2 was centered around four different behavior states, the response form was updated to include questions on state comparisons. Questions focused on perception of the system from two perspectives: relations with other players and the resulting sonification. The questions for Wave 1 (W1) were presented as follows:

1. What was your sense of playing in this virtual environment?
2. What was your sense of connection to the others in the virtual space? (Other players or agents)
3. How did you perceive your own "voice" while playing? (Location, timbre, relation to environment and others)
4. How would you describe your ability (or lack of) to perform expressive musical action?

Wave 2 (W2) introduced states which altered agent trail decay, sensory distance, "death" threshold, birth odds, and agent energy decay. Players were not primed on the behavior of each of these states. The transition between each state was announced to prompt the players that they will be interacting with new behavior. States progressed sequentially through 1-4, but could be revisited following the session. The experienced states were:

- **S1 - Solitary**: Fast trail decay, low sensory distance, default death threshold, low birth odds, and default agent energy decay
- **S2 - Needy**: Slow trail decay, low sensory distance, lower death threshold, high birth odds, and very fast agent energy decay
- **S3 - Lively**: Fast trail decay, high sensory distance, default death threshold, high birth odds, and slow agent energy decay
- **S4 - Starving**: Slow trail decay, high sensory distance, lower death threshold, low birth odds, and very fast agent energy decay

The names provided before the description of each state were given by the first author through personal interpretation of their behavior and were not told to players. Questions from W1 were all asked again, including "For each state:" before a given question. One additional question was asked:

- How would you describe the behaviours of each state? (changes in response, characteristics, etc)

Answer lengths were not prompted to be short or long, allowing players to provide as much detail as they wished. 10 player responses were recorded for both W1 and W2, and a thematic analysis was conducted on this data. Most players had little or no prior experience with participatory musical systems. A small amount had extensive prior experience with improvisational musical play.

5 Responses & Analysis

Participant responses outline a range of interpretations for *Locus Diffuse*, as various natural metaphors were attributed to the audio and visuals. One participant noted "I definitely had the sensation of being immersed in a medium – fluid. The dynamics of the particles, of course, were responsible for evoking this sensation, but so were the sounds and the way that they transformed". While players were primed that the agent behavior was emulative of slime mold, their natural metaphors for the agent behavior tended towards more commonly encountered phenomena of the natural world. Natural processes such as swarms of bugs, flowing rivers, and immersion within a fluid substance were noted as a reaction to both the aural and visual content of the simulation. While similar sources were used as granular input across both waves, the sense of a natural process was far from 1:1 with source audio, and rather was in reaction to both timbre and agent behaviour. This points to the *perception of an emergent sonic ecosystem that is influenced both by variations of the sound source and by agent behavior.*

Interaction with agents, guided by personal connection/narrative, was also a key feature of participant responses. One player noted, "There was a certain appeal to doing things like building 'bridges' between myself and other users, and seeing the cells speed up and slow down made it feel like we were almost taking care of the cells in a way". This was exemplified within states S2 (Needy) and S4 (Starving) of Wave 2, where accelerated agent energy decay and earlier death resulted in huddles of player positions protecting a core population of agents. Players attributed direct and/or implied characteristics towards agent and environmental behavior throughout each of the states. Players would alter the target of these characteristics, displaying that these changes were felt on either an agent or environmental level. Environmental-related characteristics tended to be a product of the visual aspects of the system, noting "busyness" and "growth" of agents within S2 & S4 when trail decay was reduced. For agent behavior characteristics, S1 (Solitary) was perceived as "independent", resulting in localized areas of attraction with distant agents acting indifferent to the presence of energy. One response attributed 'interest' as a quality the agents possessed, stating that "agents seem to be highly invested in the actions of players when they are sharing energy, but seem to actively avoid players who are not working together to share energy". One player noted that these states "rewarded stillness", where one's interaction felt more impactful to the sonification by waiting and allowing the agents to move towards and through them.

Audio and visual cohesion of the system was found to be necessary for players to internalize a complete understanding of the resulting agent behaviors. An interesting trend is shown in some player responses to seemingly *lean towards a visual characterization of the system state vs an aural one*. This can be seen in responses comparing S1 and S3 (Lively). Reports on each states sonic activity were contrasting, noting S1 as reserved and stable, but S3 as busy or chaotic. Although perceived as sonically contrasting, most participants noted S1 and S3 being similar due to visual qualities of the simulation, mainly trail decay rate. Similar reports occurred between S2 and S4 due to their low trail decay rate. This visual bias may also be a product of the relatively low familiarity of participants with musical systems/play experience.

Varied experiences of connectedness were reported: a lack of connection, connection mainly with the simulated agents, and connection to the meditative qualities of

communal movement. Reports of a lack of connection were attributed to a desire to have dialogue with fellow players (voice/text) in order to coordinate, or again due to a focus upon the visuals, noting "Being able to see where other players were going and patterns they were following made the connection much strong across all states". The second focus is discussed above ("interaction with agents"). The third focus was on immersion in the system and the sonification of player movements, which divided into two groups. The first noted a distinct gravitation to the mediating agent system and its behaviours, with one participant stating "I could feel each of there positions in a unique way. It was as if they were taking up space in a room". The second focused on inter-human sounding while immersed in the mediating virtual space, with one participant stating, "I found myself being more consciously aware of the other players' positions/motions, and adjusted my own motions in relation to theirs".

6 Conclusions & Future Work

Blending aspects of sonic ecosystems, agent-based musical systems, multi-user instruments, and networked performance to establish a communal musical play context, *Locus Diffuse* depicts these disparate fields of study as complimentary in their nature to establish compelling emergent behavior through various levels of interaction, sounding, group structure, and process. Employing natural computing for the mimicry of biological systems allows for flexible and dynamic collaborative musical agents by speeding up natural processes to allow them to be used in real-time musical computation tasks. The provided system overview allows for detailed understanding of agent mechanics and human/machine interplay resulting in sonification. Play sessions with *Locus Diffuse* resulted in four key observations from participant responses:

- The perception of a sonic ecosystem was tied to variation in sound sources intersected with agent behavior.
- Narrative-based personal connection between players and agents mediated interaction characteristics.
- There was a bias towards a visual understanding of the system vs an aural one.
- The "locus" of experiences of connection were more varied, ranging from a lack of systemic connection, focus on inter-human collaboration, to human-agent collaborative sounding.

Each of these outcomes is a product of the interaction between system behavior, player action, and aural & visual aesthetic decisions, constituting various networks at play between the project's amalgam of practices, communal musical goals, and telematic structure.

In a post social distancing time, an in-person room scale version of the system will be created in order to explore the translation of the current network-based musical instrument design back into the originally intended space. Translating to this physical space, perceptions related to embodied movement as a control source can be explored within the established agent-based sonic ecosystem. Further research into potential narrative outcomes of inter-human and human-agent-based collaboration may yield interesting results within sonic ecosystems. Sessions aimed at varied levels of musical experience may reveal interesting trends related to aural vs. visual attention, and attention to

inter-human or human-agent interactions. Additional telematic sessions with reduced visual feedback may also shed light on how much a purely sound-based system can express these ecosystemic interactions.

References

1. Adamatzky, A.: Physarum Machines: computers from slime mould. World Scientific, Singapore. (2010)
2. Borgo, D.: Sync or Swarm: Musical Improvisation and the Complex Dynamics of Group Creativity. LNCS, vol. 4060, pp. 1–24. (2006)
3. Carpentier, T., Noisternig, M., Warusfel, O.: Twenty Years of Ircam Spat: Looking Back, Looking Forward. ICMC. pp. 270–277. Denton. (2015)
4. Di Scipio, A.: 'Sound is the interface': from interactive to ecosystemic signal processing. Organised Sound, vol. 8(3), pp. 269-–277, United Kingdom. (2003)
5. Durham, A. C., Ridgway, E. B.: Control of chemotaxis in Physarum polycephalum. The Journal of Cell Biology, vol. 69(1), pp. 218-–223. (1976)
6. Frith, S.: Music and Identity. Questions of Cultural Identity, pp. 108–128. (1996)
7. Gale, E., Matthews, O., De Lacy Costello, B. Adamatzky, A.: Beyond Markov Chains, Towards Adaptive Memristor Network-based Music Generation. International Journal of Unconventional Computing. vol. 10. (2013)
8. Gresham-Lancaster, S.: The aesthetics and history of the hub: The effects of changing technology on network computer music. LMJ, vol. 8(1), pp. 39–44. (1998)
9. Hoy, R., Van Nort, D.: Augmentation of Sonic Meditation Practices: Resonance, Feedback and Interaction through an Ecosystemic Approach. In: Kronland-Martinet R., Ystad S., Aramaki M. (eds) Perception, Representations, Image, Sound, Music. CMMR 2019. LNCS, vol 12631. pp. 591–599, Springer, Cham. (2021)
10. Jones, J.: Characteristics of Pattern Formation and Evolution in Approximations of Physarum Transport Networks. Artificial Life, vol. 16(2), pp. 127–153. (2010)
11. Jordà, S.: Multi-user Instruments: Models, Examples and Promises. NIME, Vancouver. pp. 23–26. (2005)
12. Miranda, E. R., Adamatzky, A., Jones, J.: Sounds synthesis with slime mould of Physarum Polycephalum. Journal of Bionic Engineering, vol. 8(2), pp. 107-–113. (2011)
13. Müller, M. W.: petra for max. `http://circuitmusiclabs.com/projects/petra-for-max/` (2016)
14. Nakagaki, T., Yamada, H., Toth, A.: Intelligence: Maze-Solving by an Amoeboid Organism. Nature, vol. 407, p. 470. (2000)
15. Oliveros, P.: Deep Listening: A Composer's Sound Practice. iUniverse, Lincoln. (2005)
16. Oliveros, P., Weaver, S., Dresser, M., Pitcher, J., Braasch, J., Chafe, C.: Telematic Music: Six Perspectives. Leonardo Music Journal, p. 19. (2009)
17. Reid, C. R., Latty, T., Dussutour, A., Beekman, M.: Slime mold uses an externalized spatial "memory" to navigate in complex environments. Proceedings of the National Academy of Sciences, vol. 109(43), pp. 17490-–17494. (2012)
18. Rozenberg, G.: Handbook of Natural Computing. Springer, Berlin (2012)
19. Small, C.: Musicking: The Meanings of Performing and Listening. Wesleyan University Press, Middletown. (1998)
20. Vogel, D., Gautrais, J., Perna, A., Sumpter, D., Deneubourg, J., Dussutour, A.: Transition from Isotropic to Digitated Growth Modulates Network Formation in Physarum polycephalum. Journal of Physics D: Applied Physics, vol. 50, iss. 1. (2016)
21. Weinberg, G.: Interconnected Musical Networks - Bringing Expression and Thoughtfulness to Collaborative Group Playing. MIT, Cambridge. (2003)

Mixed Writing with Karlax and Acoustic Instruments: Interaction Strategies from Computer Music

Benjamin Lavastre[1] and Marcelo M. Wanderley[2]

[1] DCS, IDMIL, CIRMMT, McGill University
[2] IDMIL, CIRMMT, McGill University
benjamin.lavastre@mail.mcgill.ca; marcelo.wanderley@mcgill.ca

Abstract. The karlax is a gestural controller developed around 2010. Since its inception, it arose substantial interest among composers and continues to be commonly used in solo and group performances. One of the reasons for its longevity is its great adaptability especially in interaction with acoustic instruments. This article analyses six chamber music pieces for karlax and acoustic instruments by comparing the sound and visual results and the writing process (scores, patches, and mapping). We discuss different composition strategies through the use of interaction metaphors from the computer music literature. These metaphors prove to be powerful analysis tools that allow describing the use of a digital music instrument (DMI), such as the karlax, in a chamber music context.

Keywords: Mixed pieces, Computer Music, Digital Music Instruments (DMI), Electronic Chamber Music, Input Devices, Mapping

1 Introduction

Though several hundred interfaces for musical expression have been developed and described in a variety of venues, most notably in the last two decades at the International Conference on New interfaces for Musical Expression (NIME)[3], relatively few articles discuss how these interfaces ares used in actual musical contexts, for instance [1], [2], [3] and [4]. Indeed, the use of DMIs is not often discussed from the perspective of artistic and musical composition. In other words, *the "M" in NIME*: why don't we talk more about music performance with musical interfaces, beyond sound control? In part, this is the consequence that most of the interfaces described in the literature have short life spans and/or are mainly used by their designers [5]. In this sense, the karlax offers a particularly rich subject of study with an existence of more than ten years, a community of regular users from different musical cultures and several significant creations, notably with acoustical instruments, incorporating some form of music notation.

The karlax is an input device that combines several sensors: continuous keys, velocity pistons, axis, switches, and three axes of accelerometers and gyroscopes (Fig. 1) [4]. "Its ability to detect subtle as well as larger gestures, continuous as well as event-based control, its low latency and high bandwidth, its reliability and portability" has

[3] www.nime.org
[4] www.dafact.com

been praised [6]. Like many musical interfaces that output sensor data but which do not have a pre-defined sound, the karlax is defined by its control characteristics, i.e., its gestural identity instead of a given sonic identity. This opens up unlimited musical possibilities but requires the composer to describe the sounds controlled and the mapping between sensor data and sound generation to be used in each context. A digital musical instrument (DMI) is composed of the group: control interface + mapping + sound generation [7].

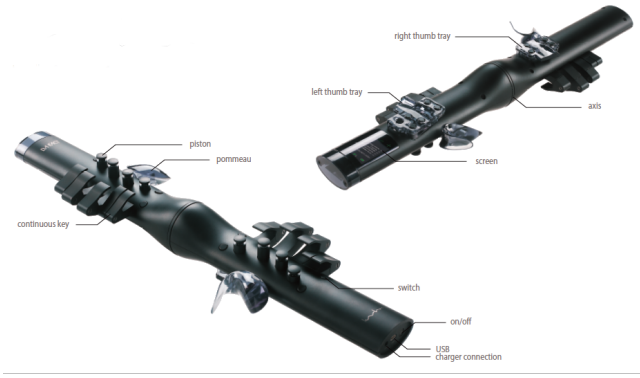


Fig. 1. Front and rear views of the karlax (www.dafact.com)

In this study, we analyze a corpus of six pieces for karlax and acoustic instruments from audio and video recordings, scores, Max/MSP patches, articles, presets, etc. We have identified three compositional models that allow us to define the main areas of inspiration for each of the pieces. In a second step, we will discuss excerpts in the corpus by detailing the action of the karlax and its interaction with the acoustic instruments thanks to interaction metaphors from Computer Music.

2 Objectives

The objectives of this article are:

1. Study of six pieces for karlax and acoustic instrument(s) including analysis of sound synthesis, mapping, gestures, and scores.

2. Among these pieces describe the ”role” of the karlax by identifying compositional models.

3. Analyse the use of the karlax and its interaction with acoustic instruments in excerpts of these pieces thanks to Computer Music metaphors.

3 Corpus of Pieces

We have selected 6 pieces written between 2013 and 2018 that combine the karlax controller with one or two acoustic instruments among a flute, a violin, and a cello. Five of the six pieces of the corpus have been commissioned by the *Fabrique Nomade* ensemble and have been performed by it. This ensemble is an "electronic chamber music ensemble that wishes to rediscover the gestures and listening of classical chamber music"[5]. In this regard, "each musician is independent and has total control over their acoustic or electronic instrument" (each instrumentalist has their own laptop and their own sound broadcasting system). This means that acoustic instruments performers trigger their own electronic part (most of the time real-time processing) thanks to a midi pedal and that the karlax cannot process in real-time the acoustic sound of an instrumentalist. This is not the case for the sixth piece where the karlax transforms the sound of the violin in real-time.

A *Fogg* by Lorenzo Bianchi for violin, cello and karlax, 2013 (performed by *Fabrique Nomade* ensemble)
B *Frottement, Bourdon, Craquement* by Francis Faber for cello, karlax and electronic, 2013 (performed by *Fabrique Nomade* ensemble)
C *Le Patch Bien Tempéré III* by Tom Mays, for flute, karlax and real time electonic, 2013 (performed by *Fabrique Nomade* ensemble)
D *Ripples Never Come Back* by Michele Tadini for violin, cello and karlax, 2013 (performed by *Fabrique Nomade* ensemble)
E *Discontinuous Devices "In-between"* by Michele Tadini for cello and karlax, 2015 (performed by *Fabrique Nomade* ensemble)
F *Le Violon, l'Oeillet et le Bambou* by Raphaël-Tristan Jouaville, for violin and karlax, 2018

4 Composition models

Among these pieces, we have identified three compositional models that represent three main sources of inspiration for the composers: model based on acoustic sounds, model based on electronic sounds and karlax as model. These allow describing the main "role" of this controller in relation to the other instruments.

Model based on acoustic instruments sounds

For several pieces in the corpus, the acoustic sound of the instrument(s) with which the karlax plays is used as the basic composition material. For example, in the piece *Fogg* (A), the sound synthesis of the karlax is realized through an additive synthesis from the spectral analysis of several violin pizzicati with different "preparations" (addition of objects like pegs attached to the string). The karlax triggers and controls processes related to the spectral content of pizzicato sounds by pressing continuous keys (control of the spectral envelope) (Fig. 2).

Other examples are pieces where the karlax plays sounds very close to the sounds played by the instrument(s) it interacts with. In this way, the acoustic instrument is

[5] www.fabriquenomade.com

"augmented" by the action of the karlax. For example, in the third part of *Discontinuous Devices* (E), the karlax activates flautando and harmonics cello samples by pressing the continuous keys. Shorter samples of the same type are also triggered by the pistons. This forms a harmonic environment for the cello, which performs more percussive figures like jettatti and glissandi that let the natural harmonics of the open strings resonate. With the same idea, in Jouaville's piece (F), the karlax plays a physical model of a string by activating the pistons in a consecutive way whose pitches are previously set up (*String Studio* module). In most of the piece, the karlax highlights and develops the melodic contour of the violin and/or creates a harmonic accompaniment (Fig. 3).

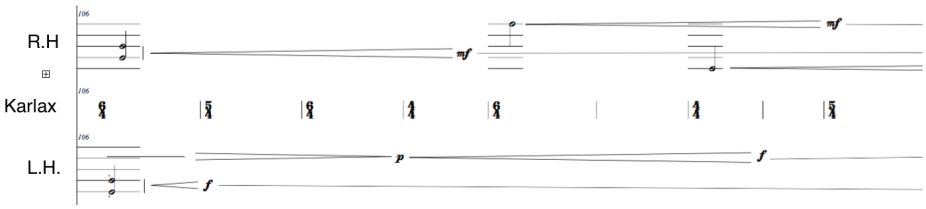


Fig. 2. "Shaping" of the spectral envelope with karlax continuous keys in *Fogg* by Lorenzo Bianchi (mes. 68-69, karlax part) (with the permission of the composer). Each staff line represents the activation of a continuous key that will control the volume of a group of oscillators.

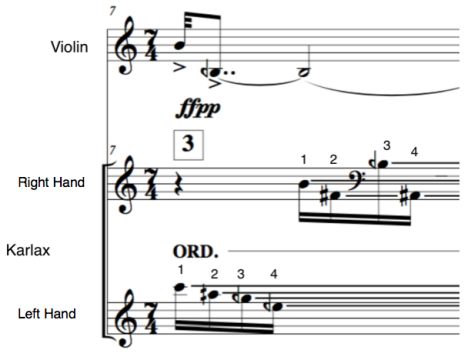


Fig. 3. Results of pitches played by the karlax pistons with the corresponding fingerings in *Le Violon, l'Oeillet et le Bambou*, by Raphaël-Tristan Jouaville (mes. 7) (with the permission of the composer). See video from 00:30 to 00:32 www.youtube.com/watch?v=IrCmiwwFSUs

Model based on electronic sounds

This type of composition model is the most common in the selected pieces. In this category are represented the treatments and manipulations associated with electronic music such as filtering, delay, granular synthesis, additive synthesis, ring modulation, arpeggiators, freeze, etc. Also, this control interface is often associated with the processing of electronic synthesis. By assigning certain parameters of the sound synthesis to

different sensors, the karlax can “drive” processes in real-time and bring an expressive dimension to the transformations. In this model, the sound of the karlax is perceived as independent from the acoustic sound of the instruments. For example, in the piece *Le Patch bien tempéré III* (C), the composer focuses on complementary electronic techniques such as harmonizers, delays, and ”paf” synthesis based on voice formants [6]. In this piece, the input device activates different synthetic voices and modifies parameters. In general, the accelerometer data corresponding to the forward, backward movements are correlated with dynamics (brightness and intensity) and the left-right movements are correlated with pitch (glissandi) while the central axis applies a speed tremolo [8]. In the score are noted the part of the flute, the karlax movements laid out on four staves, and the acoustic results (Harmonizers and Synthesis staves) (Fig. 4).

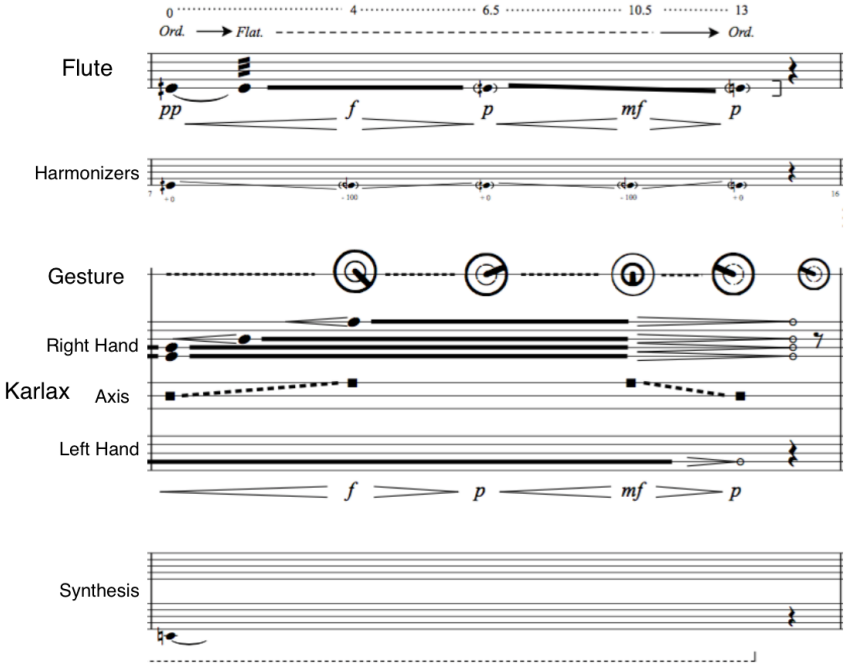


Fig. 4. General score of *Le Patch bien tempéré III* by Tom Mays (mes.6) (with the permission of the composer). The karlax part combines -movements (“Gesture” staff with circle symbols) which controls intensity, brightness, and pitch-bend of the sound synthesis, -rotation of the axis (dotted lines) which control a speed tremollo and -continuous keys depression (”Right Hand” and ”Left Hand” staves with thick lines) which activates “paf” synthesis voices. The numbers at the top of the score represent the time in seconds. See video from 01:44 to 02:00 https://vimeo.com/80464641

[6] *Phase Aligned Formant* developed by Miller Puckette in 1995

Karlax as model

The design of the karlax can also inspire the composition and constitute a model in itself. Indeed, this controller is conceived by being inspired by the keys system of wind and keyboard instruments (pistons and continuous keys) enriched with an axis (with bends) and movement sensors (accelerometer and gyroscope). The instrumental aspect of the karlax is developed among others in the introduction of the Faber's piece (B). Indeed, the instrumentalist performs a "call" thanks to the pistons produced by short harmonic synthetic sounds. The play of the karlax can be compared to the play of pistons of a trumpet (Fig. 5). Also, the possibilities of the karlax can inspire the "trajectory" of the piece. For instance, *Discontinuous Devices* (E) starts with an extensive use of the pistons and then in the second section the karlax triggers and controls long sequences through the accelerometer and gyroscope data, making the karlax gestures more and more expressive.

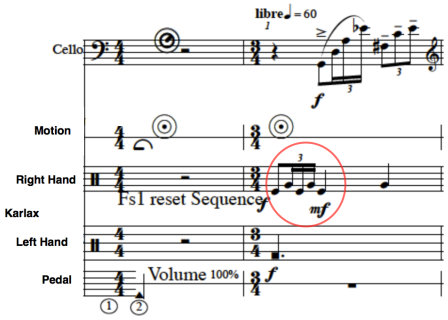


Fig. 5. "Call" played by the Karlax pistons in *Frottement, Bourdon, Craquement* (mes. 1-2) (with the composer permission). See video from 00:00 to 00:04 https://vimeo.com/118148219

5 Interaction Metaphors from Computer Music

In this part, we analyze excerpts of the corpus pieces thanks to metaphors from Computer Music. We have selected five metaphors from three articles: [9], [10], [11], for their relevance to describe the action of a gestural controller such as the karlax (particularly in interaction with acoustic instruments) and for their capacity to give an overview of compositional strategies.

"Shaping" [Caramiaux et al., 2014]

Shaping "refers to scenarios where performers control sound morphologies by "tracing" in the air those salient sound features they desire to control"[9]. This metaphor is described as the "transfer of variations into a gestural morphology" and as synchronization of sound with movement. It is widely used in the pieces thanks to Karlax motion sensors but also with continuous keys. For example, in (C), the karlax imitates the distortions of the flute sound (created by harmonizers, flatterzunge, etc.) by "shaping" the "paf" synthesis. At the same time, the ancillary gestures of the flutist seem to imitate the gestures of the controller (Fig. 4). With a more reduced gestural expression,

the continuous key activation allows the karlax performer in (A) to "shape" the spectral envelope in a differentiated way to provide a harmonic accompaniment to the violin and the cello (Fig. 2).

"Catch and Throw" [Wessel & Wright, 2002]

This strategy of interaction "involves the notion of selectively trapping musical phrases from an ongoing performance, transforming them in some way, and then sending these transformed materials back into the performance"[10]. This way of interaction, which could be defined as delayed real-time processing, is exploited in improvisational situations by Tom Mays in the early 2010's, where the direct sound of the acoustic instrument is captured, transformed by the karlax and broadcast in real-time[7]. This type of interaction is also employed at the end of Jouaville's piece (G) where the acoustic sound of the violin is processed by resonator, delay, and pitch shift modules (*GRM Tools)* whose parameter nodes are controlled by the karlax movements. This brings a sonic halo to the violin[8].

"Fishing" [Caramiaux et al., 2014]

This metaphor is related to the learning stage in gesture recognition. When a gesture is recognized by the dedicated program, a sound will be "fished" out to be played. One can compare this scenario of interactions with certain compositional strategies. For example, at the beginning of (A), several violin and cello actions with obvious gestural characteristics such as jettato, glissandi, strokes on the body of the instrument seem to be "recognized" by the karlax, which reacts by imitating gestures, triggering and transforming nearby sounds[9].

Musical tasks [Wanderley & Orio, 2002]

In the same idea as the composition model based on instrumental playing presented above (see *Karlax as model*), the article [11] proposes two levels of metaphors: *Musical Instrument Manipulation Metaphor* and *Other Metaphor*. In the first category are listed the interactions metaphors that refer to traditional instrumental playing (isolated notes, basic musical gestures like glissandi, vibrato, musical phrases, rhythmic playing, etc.) that appear for example in Faber's piece with the "call" (Fig. 5). In the second category, the authors evoke the actions of triggering of sequences but also their organization in time: synchronization, envelope control, continuous modulation features, etc.

"Space" [Wessel & Wright, 2002]

The purpose of using a control interface like karlax in this type of strategy is to "suggest musically interesting trajectories for gesture [10]". Moreover, the article emphasizes the importance of proximity and timbre in the perception of these trajectories.

[7] In this video, the karlax controls the transformations of the acoustic sound of a Sheng, a mouth-blown free reed instrument: https://www.youtube.com/watch?v=fg9TgbI4gTM

[8] See video from 05:43 to 06:42 https://www.youtube.com/watch?v=IrCmiwwFSUs

[9] See video from 00:00 to 01:10 https://vimeo.com/67049071

In addition, various strategies to suggest movements and trajectories are employed by the composers of the corpus. For example, in *Ripples Never Come Back* (D), the composer evokes a distancing through repeated sequences where the violin and cello instruments begin a quasi homorhythmic figure which is "taken up" by the electronic part performed by the karlax in the form of arpeggios towards the high register. The karlax controls a flow of notes produced by a subtractive synthesis: the axis controls the pitch of the arpeggio, the continuous keys control parameters like volume, filtering or speed while the inclination combined with a key activation controls the envelope (Fig. 6).

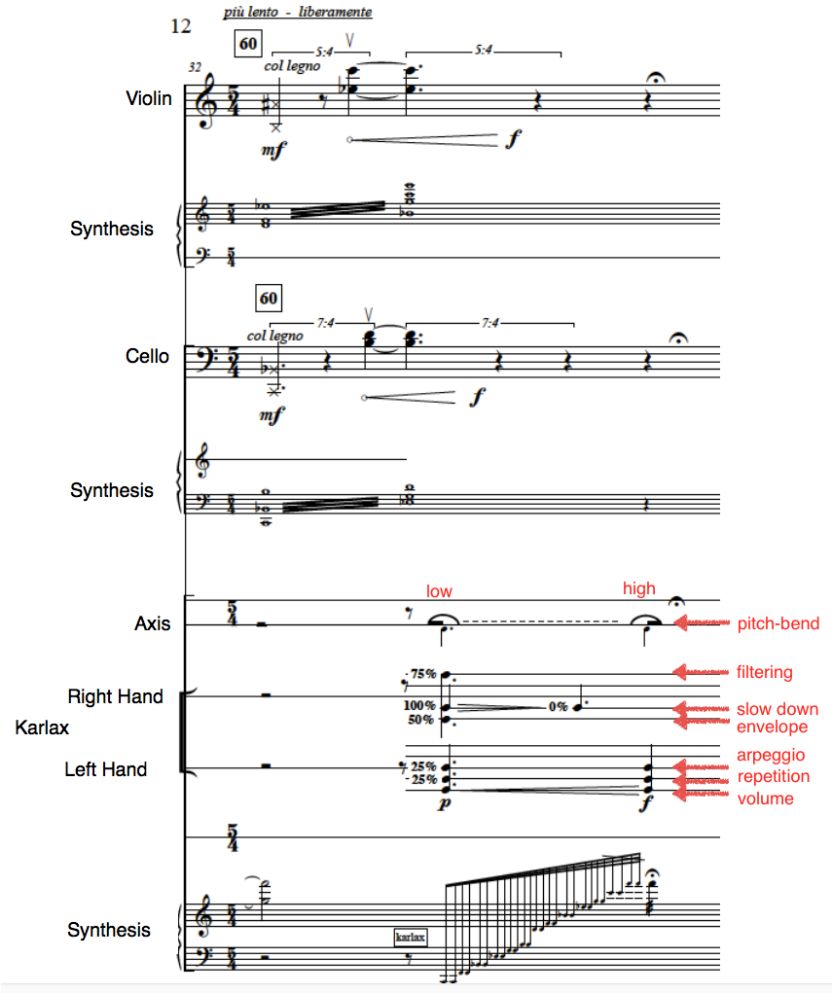


Fig. 6. Sequence that evokes a distancing in *Ripples Never Come Back* by Michele Tadini (mes. 32) (with the composer permission). See video from 00:48 to 01:00 https://vimeo.com/72995021

6 Discussion

The use of compositional models and Computer Music metaphors provide a framework and powerful analytical tools to apprehend pieces that appear at first sight very complex. It allows to categorize certain roles of the karlax in this precise context, with a small number of acoustic instruments, and allows to discuss situations.

For example, the piece (A) seemed to us to belong to both the first and the second composition model, depending on whether one considers the process of composition or the sound result. Indeed, the process of additive synthesis and the fact that the "target" sounds are prepared (with the addition of pegs) make the sound synthesis played by the karlax particularly distant from the acoustic sound of the violin. From a perceptual perspective, we would then need to determine whether or not the timbre of the sounds played by the karlax "blends" with the sound of the instrument and determine what allows us to assert this. For the other examples given for the first model: (E) and (F), we can use the terminology of "timbral augmentation" as presented in [12].

The selected metaphors are thought in real-time interactions context. While the composition process necessarily evolving in a delayed time, we have seen that these metaphors are proper to comment on typical situations of the pieces of the corpus. Firstly, because they offer situations of real-time transformations and secondly because the composition strategies in terms of dramaturgy can be compared to situations of improvisations. Moreover, the setup chosen by the *Fabrique Nomade* ensemble influences these strategies. As the instrumentalists are independent and trigger more or less random processes (for example delays), the composer tends to opt for "encompassing" strategies, highly describable by the metaphors [13]. On the other hand, these metaphors are limited to comment precisely on temporal and rhythmic aspects as specified in the article [11]. In addition, metaphors that qualify the action of a controller such as *Shaping*, or *Musical Tasks* facilitate the interaction with the instrumentalist(s) and the "reading" of the piece by the spectator/listener as they help to identify acoustically and gesturally the part played by the karlax.

Another important aspect to qualify the action of the karlax is its notation. Depending on the project of each piece, composers adopt a prescriptive (oriented on the action of the karlax player) and/or descriptive method of notation (which reports the acoustic result)[14]. As a reference point, the composers of the corpus use the basics of karlax notation presented in the article [6]. We can mention however the more pragmatic approach described in the Jouaville's piece (G) which consists in assigning events in order of appearance to a simple range of fingerings and allows to visualize the pitches played by the karlax and movements on a single staff (Fig. 3). Also, it is particularly interesting to relate the approach of the composer Andrew Stewart notably in his piece *Ritual* (2015) for karlax solo, based among others on gestures categorization and a spatial representation of space in the form of a grid [15]. In general, composers add rarely information related to mapping and sound synthesis, which would allow performers to further appropriate the karlax instrument. Simultaneously, the notation must be practical and represent the composer's intention in a precise and concise way. As such, an indication in the score of the metaphorical context, as presented above, would provide valuable information about the way(s) the karlax is played and how it interacts with other instrument(s).

7 Conclusions

In this article we presented an analysis of six pieces for karlax and acoustic instruments. Three models of compositions have been identified and five metaphors from Computer Music have been proposed to characterize typical musical situations. To go further, it seems particularly interesting to deepen the analysis of these pieces by providing a detailed description of their conception and by comparing them both in terms of sound synthesis, mapping, gestures, notation, and interactions. In addition, it would be interesting to compare the use of the karlax with other DMIs like T-Stick in the same type chamber music context.

Acknowledgments. The authors would like to warmly thank Rémi Dury, Francis Faber, Tom Mays, Andrew Stewart, Michele Tadini, Lorenzo Bianchi, Raphaël-Tristan Jouaville and Richard McKenzie for sharing their resources and their time.

References

1. Dobrian, C., Koppelman, D.: The E in NIME: Musical Expression with New Computer Interfaces. *Proc. Int. Conf. on New Interfaces for Musical Expression*, pp. 277–282 (2006)
2. Palacio-Quintin, C.: Eight Years of Practice on the Hyper-Flute : Technological and Musical Perspectives. *Proc. Int. Conf. on New Interfaces for Musical Expression*, pp. 293–298 (2008)
3. Morreale, F., McPherson, A.: Design for Longevity: Ongoing Use of Instruments from NIME 2010-14. *Proc. Int. Conf. on New Interfaces for Musical Expression*, pp. 192–197 (2017)
4. Hödl, O.: 'Blending Dimensions' when Composing for DMI and Symphonic Orchestra. *Proc. Int. Conf. on New Interfaces for Musical Expression*, pp. 198–203 (2019)
5. Ferguson, S., Wanderley, M. M.: The McGill Digital Orchestra: An Interdisciplinary Project on Digital Musical Instruments. *Journal of Interdisciplinary Music Studies* (2010)
6. Mays, T., Faber, F.: A Notation System for the Karlax Controller. *Proc. Int. Conf. on New Interfaces for Musical Expression*, pp. 553-556 (July, 2014)
7. Miranda, E., Wanderley, M.: *New Digital Musical Instruments: Control and Interaction Beyond the Keyboard*. A-R Editions, Inc. (2006)
8. Mays, T.: L'Harmoniseur augmenté : Le dispositif d'écriture mixte dans l'oeuvre *Le Patch Bien Tempéré III*". *Proc. Journées d'Informatique Musicale* (July 2015)
9. Caramiaux, B., Françoise, J., Scnell, N., Bevilacqua F.:Mapping Through Listening. *Computer Music Journal*, 38(3):34-48 (Fall, 2014)
10. Wessel, D., Wright, M.: Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*, 26(3):11-22 (Fall, 2002)
11. Wanderley, M., Orio, N.: Evaluation of Input Devices for Musical Expression: Borrowing Tools from HCI. *Computer Music Journal*, 26(3):62-76 (Fall, 2002)
12. Touizrar, M., McAdams, S.: Aspects perceptifs de l'orchestration dans *Angel of Death* de Roger Reynolds: Timbre et groupement auditif. Dans P. Lalitte (Ed.), *Musique et cognition : Perspectives pour l'analyse et la performance musicales*, (pp. 55-78). Editions universitaires de Dijon (2019)
13. Dahl, L., Wang, Ge.: Sound Bounce: Physical Metaphors in Designing Mobile Music Performance *Proc. Int. Conf. on New Interfaces for Musical Expression*, 15-18 (June, 2010)
14. Kanno, M.: Prescriptive notation: Limits and challenges. *Contemporary Music Review*, 26(2):231-254 (2007)
15. Stewart, A.: Karlax Performance Techniques: It Feels Like... *Proc. Int. Computer Music Conference* (2016)

3D skeleton motion generation of double bass from musical score

Takeru Shirai and Shinji Sako

Nagoya Institute of Technology
clf19021@ict.nitech.ac.jp

Abstract. In this study, we propose a method for generating 3D skeleton motions of a double bass player from musical score information using a 2-layer LSTM network. Since there is no suitable dataset for this study, we have created a new motion dataset with actual double bass performance. The contribution of this paper is to show the effect of combining bowing and fingering information in the generation of performance motion, and to examine the effective model structure in performance generation. Both objective and subjective evaluations showed that the accuracy of generating performance motion for double bass can be improved using two types of additional information (bowing, fingering information) and improved by constructing a model that takes into account bowing and fingering.

Keywords: LSTM network, Performance motion generation, 3D model, Double bass

1 Introduction

Double bass plays an important role as the foundation in various forms of ensemble music such as orchestral music, chamber music, wind music, and jazz. In addition, the double bass plays a solo role while accompanied by the piano or orchestra. In the case of the bowed stringed instrument to which the contrabass belongs, there is so much visual information that the timing of the sound can be shared among the players by the motion of the right arm, and the pitch can be estimated by the shifting and fingering of the left hand.

In an actual ensemble performance, visual information is an important element for conveying performance timing and specific musical expressions to other players and for facilitating ensemble performance [1]. In particular, visual information is considered to be highly important in situations where many people are playing together in an ensemble, such as in an orchestra or wind band.

In spite of the fact that visual information is one of the most important elements in playing music as described above, among the major study fields of music information processing, the studies on automatic performance generation (i.e. performance rendering) mainly focus on performance sounds, and only a few studies focus on visual information of performances.

Therefore, in this study, we aim to generate performance motion for the double bass. There are two major technical issues: the generation of natural playing motion and the

naturalness of the 3D model appearance and rendering accuracy. In this study, we first focus on the former, which is the more essential issue.

There have been some studies on automatic performance generation focusing on visual information, but they have targeted piano [2] and violin [3, 4], and generated performance motion using actual performance sounds or MIDI as input data. In the case of a bowed stringed instrument, it is considered that it is difficult to generate the performance motion such as bowing and fingering from the pitch information because motion is not uniquely determined from the pitch information. In order to solve this problem, we propose a method to generate performance motion using the musical score information that includes not only pitch information, but also bowing and fingering information that greatly affects performance motion. Some methods have been proposed for automatically estimating fingering from musical score [5], and a method combining these studies would be promising, but in this study, we suppose these additional information are added manually.

Since there is no suitable dataset for this study, it is necessary to construct a dataset of musical scores and 3D motions. In previous studies, joint points extracted by using body tracking technology of video data were used as motion information [2–4]. This approach is also superior in that it does not interfere with the playing motion. However, in this study, which targets a large instrument such as a double bass, it is considered difficult to obtain accurate performance motion using this technique because part of the performer is hidden by the instrument. Therefore, we collect motion data using the inertial motion capturing device.

In this study, we adopted LSTM (Long Short Term Memory) network as a model for the conversion between musical score data and motion data. In particular, we verify the effect of using additional information (bowing and fingering information) as input data by comparing the accuracy of the generated motion only from pitch information and with additional information. Furthermore, we design a series model that learns the right arm motion from the bowing information and the left arm motion from the fingering information independently, and verify the effect of changing the structure of the model.

2 Related works

Li et al. [2] proposed a method to generate a pianist's 2D motion from MIDI sound sources of a piano performance. They used a Convolutional Neural Network (CNN) to extract the stream of the piano performance and the features of the beat structure, and used these as input data to the 2-layer LSTM network, and used the 2D performance motion from a fixed position as output data. In the subjective experiment, no significant difference was found between the human motion and the generated motion in 75% of the songs, indicating that the system does not generate extremely unnatural motion.

Liu et al. [3] proposed a method for generating violinist's performance motion from actual performance sounds. In this method, a model for predicting the bowing of the right arm and a model for predicting the expressive motions of the whole body were constructed from the Mel-spectrogram[1] obtained by performing STFT (Short Time

[1] Spectrogram in the Mel scale, a perceptual measure of pitch in human hearing.

Fourier Transform) on the input sound source. And a model for predicting the position, fingering, and strings of the left arm from the data obtained by pitch detection is constructed independently. In addition, a model that predicts the position of the left arm, fingering, and strings based on the data obtained from pitch detection was constructed independently, thereby realizing the generation of violinist's full-body performance motions.

3 Proposed method

Our model is based on a previous study by Li et al. [2]. The difference between our model and the previous study is that the output data is not 2-dimensional but 3-dimensional, and the input is not derived from MIDI but from score information, which is a sequence of symbols. We need to consider a model that addresses these differences.

We construct a 2-layer LSTM network, and use MAE (Mean Absolute Error) as the loss function and Adam [6] as the optimize function. The output vectors of the LSTM network are fed to all the coupling layers to obtain the positional and rotational information of each joint point in each frame in 6 dimensions.

We also attempt to apply the framework constructed by Liu et al. [3]which consists of three models: a bowing model for the right arm, a position model for the left arm, and a representation model for the upper body. In this study, since we are trying to generate performance motion using manually additional information (bowing, fingering) rather than performance sound data, we can treat these information as more accurate and reliable than that obtained by estimation.

Extract the sequence of pitch, bowing, and position from the musical score information as shown in the Fig. 1 into a format that can be input to the LSTM network, each with the same period. As a result, the pitch sequence is a 30-dimensional sequence consisting of $\{E0, F0, \ldots, A3\}$, the bowing sequence is a 2-dimensional sequence consisting of $\{down\text{-}bow, up\text{-}bow\}$, and the position sequence representing fingering information is a 12-dimensional sequence consisting of $\{0, 1, \ldots, 11\}$.

The three sequences extracted from the above score information are used as input data, and the sequence representing body motions are used as output data to construct a model. The goal is to verify the significance of each data and to design a model that is suitable for learning. By comparing the accuracy of the generated performance motion by the designed models, we can verify whether the bowing and position information used as additional information are significant in improving the accuracy of the generated performance motion.

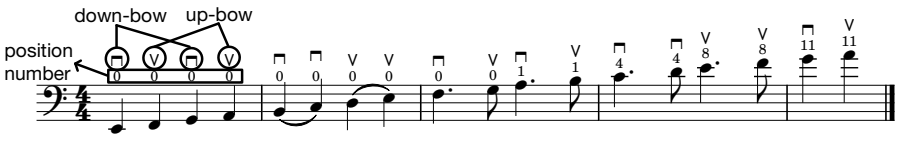


Fig. 1: Sample of musical score

	input	output
Model1	pitch(30-d)	upper body(90-d)
Model2	bowing+position+pitch(44-d)	upper body(90-d)
Model3	bowing(2-d), position(12-d), pitch(30-d)	right arm(24-d), left arm(24-d), other(42-d)

Table 1: Structure of the three models

The structure of the three models we designed is summarized in the Table 1.

4 Experiment

4.1 Dataset

We use ten pieces from the collection of exercises "Franz Simandl / 30 Etudes for the Double Bass" from No.1 to No.10 for the training data, and three pieces from No.11 to No.13 for the test data. The total time to play these 13 pieces at the tempo specified in the score is about 30 minutes.

Motion data We use an inertial motion capture PERCEPTION NEURON made by NOITOM to construct a dataset of the performance motion of one male double bass player. The bvh file is a motion capture data file format proposed by Biovision, and consists of two parts: a hierarchy part describing the tree structure of each joint point, and a motion part describing the motion data. In this study, the hierarchy part was defined as the 15 joint points of the upper body with the hip as the parent node. And since the motion part describes the position information and rotation information of each of the 15 joint points, it is represented by a 90-dimensional sequence. In this dataset, the coordinates of the parent node are set to the origin.

Since the experiment was intended to be performed at the tempo dictated by the musical score, we recorded the music performance played to a metronome. The frame rate was set to 30 fps in accordance with previous research [4]. Since the accelerometers at each joint point may deviate from their default positions due to motion during performance, calibration (correction of deviations in sensor position information) was performed after each etude was recorded.

Musical score data The musical score data was authorized for the target etudes in MusicXML format [7] using the score authoring software MuseScore. Since this study does not target the generation of expressive motion, we exclude tempo changes, volume marks, and detailed articulation instructions such as tenuto and staccato from the authoring.

In the original score, there is no bowing and fingering information for all notes, so we added symbols as bowing information and position numbers as fingering information, as shown in the Fig. 1. The position number in this case is not the actual position where the string is pressed, but the position of the index finger when pressing the string,

and is defined as position number($= \{0, 1, \cdots, 10, 11\}$), starting with the lowest pitch position.

Extract a 30-dimensional pitch sequence, a 2-dimensional bowing sequence, and a 12-dimensional position sequence for every etude in order to get the pitch, bowing, and position information from the xml data into a format that can be input to LSTM network. In this process, each information was extracted at 30 fps to match the frame rate of the motion data.

4.2 Objective evaluation

For objective evaluation, we compare the generated data with correct data (motion data collected under the same conditions as when the data set was constructed), using the following two criteria.

1. Average of the difference of coordinates at each joint point in each frame

2. Average of the ratio of the change between adjacent frame at each joint point

In the criterion 1, accuracy is verified by the difference of the amount of motion of all joint points, so the smaller the value, the higher the accuracy. The criterion 2 takes into account the problem that the only evaluation based on the criterion 1 is not sufficient because of the not so small difference in the amount of motion among joints. The criterion 2 verifies the accuracy by the ratio, so the closer the value is to 1, the higher the accuracy.

The results for the criterion 1 are shown in Fig. 2(a), and the results for the criterion 2 are shown in Fig. 2(b). From these two figures, it can be seen that the order of accuracy is Model1 $<$ Model2 $<$ Model3.

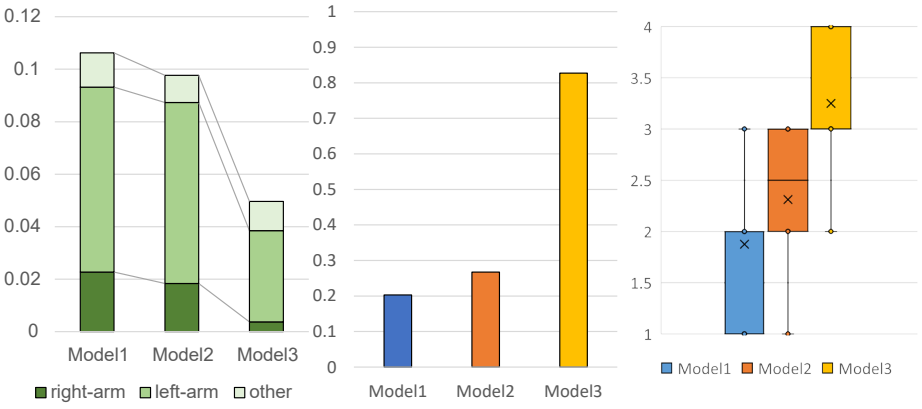
4.3 Subjective evaluation

The subjective evaluation is based on the naturalness of the performance motion. In order to make this evaluation, it is necessary to have a person who can concretely imagine the performance motions of the player from the score information, so the subjects of the evaluation experiment were limited to double bass players. After checking the score, the subjects watched a movie of the generated motion data played on Blender. In this evaluation experiment, the order of playback was randomized. A total of 16 double bass players, 8 males and 8 females in their early 20s, evaluated the naturalness in four levels: “1: unnatural”, “2: somewhat unnatural”, “3: somewhat natural” and “4: natural”.

From Fig. 2(c),which show the results of subjective evaluation using a box-and-whisker diagram, it can be seen that the order of accuracy is Model1 $<$ Model2 $<$ Model3. This is consistent with the result of the objective evaluation.

5 Conclusion

In this study, we proposed a method for generating performance motions of a double bass, for which it is difficult to predict performance motions from audio signals, by using musical score information (pitch, bowing, and fingering) as input data. As a result



(a) Objective evaluation(1) **(b)** Objective evaluation(2) **(c)** Subjective evaluation

Fig. 2: Results of evaluation

of the experiments, it was demonstrated that there was a positive effect of providing additional information (bowing and fingering), and that a higher effect could be obtained by learning the right arm and the left arm independently from the bowing and fingering information. As a future task, the generation of expressive performance motion is considered. In addition, the generation of realistic performance motions using 3D human models will be useful for performance training for beginners.

References

1. Satoshi Kawase. Communication between ensemble performers: Coodination cues. *Japanese psychological review*, Vol. 57, No. 4, pp. 495–510, 2014. (in Japanese).
2. Bochen Li, Akira Maezawa, and Zhiyao Duan. Skeleton Plays Piano: Online Generation of Pianist Body Movements from MIDI Performance. In *International Society for Music Information Retrieval*, pp. 218–224, 2018.
3. Jun-Wei Liu, Hung-Yi Lin, Yu-Fen Huang, Hsuan-Kai Kao, and Li Su. Body Movement Generation for Expressive Violin Performance Applying Neural Networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3787–3791. IEEE, 2020.
4. Eli Shlizerman, Lucio Dery, Hayden Schoen, and Ira Kemelmacher-Shlizerman. Audio to body dynamics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7574–7583, 2018.
5. Wakana Nagata, Shinji Sako, and Tadashi Kitamura. Violin fingering estimation according to skill level based on hidden markov model. In *International Computer Music Conference*, 2014.
6. Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
7. Michael Good. MusicXML: An internet-friendly format for sheet music. In *Xml conference and expo*, pp. 03–04. Citeseer, 2001.

Lyric document embeddings for music tagging

Matt McVicar, Bruno Di Giorgi, Baris Dundar, and Matthias Mauch

Apple
mmcvicar@apple.com

Abstract. We present an empirical study on embedding the lyrics of a song into a fixed-dimensional feature for the purpose of music tagging. Five methods of computing token-level and four methods of computing document-level representations are trained on an industrial-scale dataset of tens of millions of songs. We compare simple averaging of pretrained embeddings to modern recurrent and attention-based neural architectures. Evaluating on a wide range of tagging tasks such as genre classification, explicit content identification and era detection, we find that averaging word embeddings outperform more complex architectures in many downstream metrics.

Keywords: lyrics, word2vec, doc2vec, music tagging

1 Introduction

Song lyrics have been shown to be effective predictors of emotion [31], and can be indicative of genre [25, 3, 32, 15, 16, 14, 13], mood [7, 32, 6, 8, 11, 2], music exploration [29], song structure analysis [28] and other musical facets such as quality and release date [22, 19, 3]. This makes them good candidate features for automatic music tagging (assigning labels like *pop*, *chill* to songs).

In the literature Hu and Downie [7] use collections of n-gram word counts (along with audio) for classifying mood. Mayer et al. [14] classify genre via rhyme analysis of lyrics, and Van Zaanen and Kanters [26] re-weight the word counts using TF-IDF (see 2.2) to classify musical moods. Text in these studies is often represented in *Bag of Words* format [7, 14, 15], where a vocabulary is built from a corpus and a song is represented as counts of the corpus words [5]. To obtain a usable vocabulary size, words are typically removed from the corpus if they appeared too often (stopwords such as *the*, *a*) or not often enough (bespoke vocabulary and misspellings).

Bag of Words is a useful intuitive document representation, but does not account for the fact that some words may have a low count in a document, yet still be considered interesting from a corpus perspective (for example, the word *algorithm* in a corpus of agricultural documents). Term Frequency Inverse Document Frequency (TF-IDF) [9] accounts for this by multiplying Bag of Words by a factor representing how common a word is in a corpus, and has also been explored in the music tagging context [26].

The methods above have some clear drawbacks. First, no semantic meaning is preserved or inferred between the individual words, meaning for example the model shares no information between words such as *love* and *adore*. Second, the feature vectors can also easily become large and sparse (due to large vocabularies), making their use in

machine learning models unwieldy. Word2vec [17] mitigates both of these issues by learning dense representations from a corpus, i.e. each word is represented as a point in a low-dimensional space in which semantically similar words are close. The training objective in this model is to predict either a missing word given the context (Continuous Bag of Words) or vice-versa (Skipgram). Word2vec has been adopted in a wide range of NLP tasks, including machine translation [24], sentiment analysis [33], and text generation [1]; and in the Music Information Retrieval (MIR) domain has successfully been applied to explicit song detection [20], genre classification [10] and music recommendation [27].

Although word order is considered in word2vec training, the algorithm does not provide a method for representing a document - something which is often needed for downstream tasks [3, 22]. One solution is to take summary statistics of the constituent word embeddings (i.e. simple averaging [27]). Another approach from the NLP literature is Doc2Vec [12], which learns paragraph-level representations of documents via an additional model input representing paragraph indices. Finally, it is possible to train the aggregation of word into document embeddings, for example using the final state of a recurrent neural network or the output of a self-attentive probe layer [30]. Advanced models such as these were used by Alexandros Tsaptsinos [25] to classify 20 music genres in a corpus of around 500,000 documents.

Solving these two problems (large vocabulary size and variable sequence lengths) is crucial to designing an accurate music tagging system from lyrics. Making this work practically, and at scale, is the subject of this paper. More concretely we investigate the efficaciousness off "off-the-shelf" language models trained on $\mathcal{O}(100B)$ tokens, training our own word embeddings from scratch on a bespoke lyrics dataset, and "warm-starting" the training. To produce a representation of an entire song, we evaluate whether word-level features should be averaged, or processed using recurrent architectures. It is out hope that this paper will serve as a practical guide for researchers hoping to make use of lyrics in tagging tasks.

2 Methods

The core of our investigation is trialling several options of representing song lyrics as an embedding. For this purpose we chose a transfer learning setup with distinct document embedding and tagging stages (Figure 1). This setup has benefits beyond our investigation: the document representation can be learned from massive amounts of unlabelled lyrics, and can be re-used for different downstream tasks. We describe the model components in detail below, beginning with some definitions.

2.1 Definitions

In line with the NLP literature, we will refer to the *lyrics to a song* as a *document*, and to a *collection of lyrics* as a *corpus*. A document is made up of multiple *words*, usually broken by whitespace, but it is sometimes more convenient to work with subword *tokens* so that information can be shared between words like *play*, *played*, and *playing* in a model. Broader structural information within documents come in the form of *sentences*

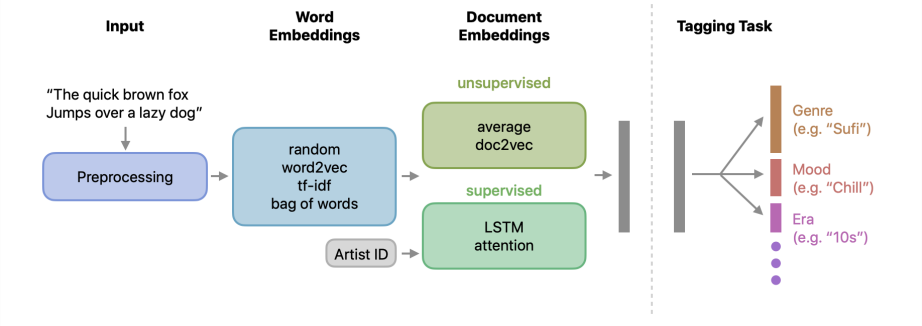


Fig. 1: We begin by processing raw lyric strings, before generating embeddings at the word level. We then have various methods of computing document embeddings: both supervised (sequence models with artist identification as the target) and unsupervised (averaging and doc2vec). At the end of this process we have a single embedding per document, which is our proposed representation. We then evaluate these embeddings by training deep tagging classifiers on the same representation.

(groups of words separated by a period or other punctuation) or *paragraphs* (longer groups of words separated by line breaks). Lyrics do not often feature well-defined sentences but instead are arranged into *lines* and *stanzas*. As these are roughly analogous to sentences/paragraphs, so we will refer to them as such in the remainder of the paper.

2.2 Word embeddings

Baseline Models We begin by defining some simple baseline models:

- `random`: random embeddings of dimension 128.
- `bag-of-words-d`: bag-of-words models with dimension d.
- `tf-idf-d`: TF-IDF models of dimension d.

For `bag-of-words-d`/`tf-idf-d`, we trimmed the vocabulary of the corpus by removing words which appeared in at least 90% of documents, and then retained the d most commonly occurring words. We had initially planned to reduce the dimensionality of the baseline models in a more principled way through dimensionality reduction techniques such as Principal Component Analysis, but realized that even with a sparse implementation we could not scale these techniques to our dataset size.

Custom-trained word2vec Next, we trained word2vec models on our dataset, using the Python package `gensim`[1] to omit words which occurred fewer than five times in the dataset, and trained for 5 epochs – these hyperparameters seemed sensible enough that we did not attempt to optimize them. We did however try several embedding dimensions for use in downstream evaluation (see Table 2), and refer to these models as `word2vec-d` for dimensionality d.

[1] `https://radimrehurek.com/gensim/`

Pre-trained word2vec The appeal of pretrained embeddings is that they have been exposed to a massive amount of text – typically several orders of magnitude larger than in-house datasets. They can therefore learn a good general-purpose understanding of word semantics, which can then optionally be fine-tuned on a specific domain task. For our experiments we used the google news 300 dataset, which contains 300-dimensional vectors for around 3 million words, trained on around 100 Billion tokens [17]. Naturally some words appeared in our data for which no pretrained embedding existed - these were simply omitted. We refer to this model as `google-300`.

Warm-start word2vec We also attempted to "warm-start" the training of the embeddings from the model above into new embeddings `google-300-warm` - these vectors retained their dimensionality and we kept the same training hyper-parameters as `word2vec-d`. The vocabularies of the two models were merged, such that words which appeared in both models took their initial state from `google-300` whilst words which were unique to `word2vec-d` had random initial state.

2.3 Word Embedding Summaries

For all representations above, to obtain a document-level representation we used averaging (for `word2vec-d`) or the native summary statistic (e.g. summing word counts in a document for `bag-of-words-d`).

In order to take paragraph structure and/or word order into account when computing document embeddings, we make use of more sophisticated summarization techniques. This section investigates various methods for achieving this.

doc2vec We begin with doc2vec [12], once again using the `gensim` implementation. We refer to these models as `doc2vec-d`.

LSTM and Attention Next, we kept the best-performing word embeddings from Subsection 2.2 and experimented with two neural sequence models: Long Short Term Memory networks (`lstm`), and an attention network (`attention`). In order to learn the sequence parameters for these models, we needed a target for the model to predict. Not wanting to use any labels which would be later used in our evaluation framework (see 2.4), we decided to use the artist identifier as the target.

The number of unique artists in our dataset is naturally very large, so we considered using negative sampling [18] to simplify the task for the networks. However, we noticed in prior informal experiments that good results can actually be obtained with a large softmax layer instead. Practically speaking, we proceeded by selecting the $1,000$ most common artists in the dataset and computing their song counts. We then randomly sampled as many songs as we could for each of these artists such that we obtained a balanced dataset. The final state for `lstm`, or the aggregated embedding for `attention`, were then connected to the target with dense layers.

We defer the discussion of results until Section 4, but note here that both these models achieved a categorical accuracy in the artist identification proxy task of around

0.85[2]. In the next Subsection we describe our tagging model and the datasets used to evaluate and compare different document embeddings.

2.4 Tagging Framework

Multi-label Our tagging model is a multi-task neural network architecture, where predictions on different tag vocabularies are treated as different tasks. The input embedding is projected through a stack of fully connected layers until it branches to a number of linear output layers, one per tag vocabulary. The loss function used to train the network is obtained by summing the binary cross-entropy loss terms associated with the output branches. Note that binary cross-entropy loss is used instead of the categorical cross-entropy loss because multiple tags within the same vocabulary can be active for the same document.

Multi-task We use multiple annotated datasets, defined over different set of documents: each dataset defines its own tag vocabulary and task. The multi-task formulation makes it convenient to handle missing annotations, while still training all tasks in parallel. The overall loss is:

$$\mathcal{L}_d = \sum_i \lambda_i a_{i,d} \mathcal{L}_{i,d}, \tag{1}$$

where $\mathcal{L}_{i,d}$ is the loss term associated with the i-th task for document d, λ_i the loss weight for the task, and $a_{i,d} \in [0, 1]$ a binary flag that represents whether document d is present in the annotations for task i. When a track does not appear in an annotation dataset, the loss terms associated with that dataset is set to zero.

Training During training, mean Average Precision (mAP) is computed at each epoch, and training is stopped when mAP reaches a plateau on the validation set. Vocabulary-wise metrics are obtained simply by averaging the values for each tag, and a final scalar value is obtained by averaging across all tag vocabularies, weighted by the number of tags in each vocabulary. A summary of the hyper-parameters searched for all models is shown in Table 2.

3 Datasets

Lyrics datasets We began with an internal dataset of 17,389,303 documents with primary language as English.[3] Documents were then tokenized in `gensim` via the `simple_preprocess` function. We discovered that the distribution of number of tokens in the documents had an extremely long tail. This was prohibitive for sequence models, so for all document embedding experiments we truncated the number of tokens to 512, which reduced the maximum sequence length from 8,641 to 512 yet only affected 4% of documents. After preprocessing, we were left with a corpus of approximately 3.8 billion tokens.

[2] a random classifier would score around 0.001

[3] deriving multiple language embeddings is an interesting extension of our work but beyond the scope of this paper

Tagging datasets We trained the tagging models on a set of internal datasets that were either manually curated or created from metadata. The datasets contain tags from different domains, e.g. genre, mood, release date, and are defined over different, but overlapping, set of documents. A description of the datasets is provided in Table 1.

					Examples per label		
Dataset	Example tag	Tracks	Tags	Tags/track	Min	Mean	Max
Flagger	`spoken`	58,444	6	1	2,700	9,740	39,796
Era	`10s`	50,450	9	1	87	5,769	17,616
Moods	`Chill`	71,271	22	1.9	100	6,092	21,995
Explicit	`True`	48,683	2	1	16,234	24,241	32,449
Genre-1	`Sufi`	2,702,226	460	2.1	25	5,835	122,224
Genre-2	`Piano`	479,792	273	1	490	1,757	2,000
Genre-3	`East Coast Rap`	39,087	261	3.3	21	497	11,105
Genre-4	`Worship`	562,274	25	3.6	152	79,966	426,008

Table 1: Tag datasets used for evaluation. Tags/track is simply the number of tags per track, averaged over each dataset.

Some of the tag datasets may contain multiple labels for the same track, which makes creating balanced data splits more challenging. We used iterative splitting [21], while also forcing tracks from the same album to appear in the same split [4]. Note that some of the tagging datasets do overlap with the datasets used to train the document embeddings. However the risk of overfitting here is small because the only label we use for training our embeddings is the artist identifier (see Subsection 2.2).

4 Results

4.1 Word embeddings

We show our results for overall mAP using word embeddings in Figure 2, showing only the best-performing model dimensionality in each group. All models outperform the random baseline but accuracy is varied across the tasks, owing in part to the differences in vocabulary size (recall Table 1). The `word2vec-512` model with averaging achieves top performance on 6 of the 8 tasks, and is a close second on the `Flagger` task.

The only task on which a pretrained model is able to compete with `word2vec-512` is on the `Moods` dataset. In general, warm-starting the training of embeddings did not yield improvements on our evaluation datasets.

4.2 Document embeddings

We selected `word2vec-512` as our best-performing word-level embedder, and set out to see if we could improve over simple embedding averaging – see Table 3 for our

Hyperparameter	Values
embedding dimension	$\{2^7, \ldots, 2^9\}$
dropout	$\{0.1, \ldots, 0.9\}$
learning rate	$\{10^{-1}, \ldots, 10^{-5}\}$
dense layers	$\{2^0, \ldots, 2^3\}$
dense size	$\{2^3, \ldots, 2^9\}$
lstm units	$\{2^6, \ldots, 2^9\}$
attention probes	$\{2^2, \ldots, 2^5\}$
attention mapping dimension	$\{2^2, \ldots, 2^5\}$

Table 2: Hyper-parameters evaluated in our experiments. Bayesian hyperparameter optimization [23] was used to optimize the validation mean Average Precision, with early stopping and patience of 10 epochs. 20 trials were run concurrently and in total 100 trials were conducted for each model.

results. Here we see that only `attention` is able to compete with `word2vec-512`, reaching similar performance on `Genre-3` and superior scores on the `Moods` and `Explicit` datasets.

Given the ability of `attention` to effectively label moods and explicit content, it seems that artist identification was a suitable proxy task for training the sequence models, or that the attention architecture is well suited for tasks related with specific keywords, such as emotions for moods or offensive content for `Explicit`.

It is unclear why the powerful `lstm/attention` models do not yield higher scores. One reason could be that we have sufficient data to train excellent word embeddings, such that further refinements are simply hard to realize. With this in mind, and knowing that in many cases large amounts of data are difficult to come by, we were interested to see what kind of performance could be attained from subsets of our data.

	Flagger	Era	Moods	Explicit	Genre-1	Genre-2	Genre-3	Genre-4
word2vec-512	**0.429**	**0.365**	0.202	0.687	**0.086**	**0.065**	0.095	**0.366**
doc2vec-512	0.368	0.271	0.183	0.727	0.060	0.037	0.069	0.358
lstm	0.330	0.247	0.204	0.723	0.044	0.041	0.057	0.282
attention	0.427	0.295	**0.272**	**0.760**	0.070	0.057	**0.107**	0.350

Table 3: Mean average precision for each model and tagging dataset for computing document embeddings. Best results for each dataset are in boldface.

4.3 Incremental training

We trained `word2vec-512` on random subsets of our data: 0.001%, 0.01%, 0.1%, 1%, 10%, retaining the full evaluation test set in each case. Results can be seen in

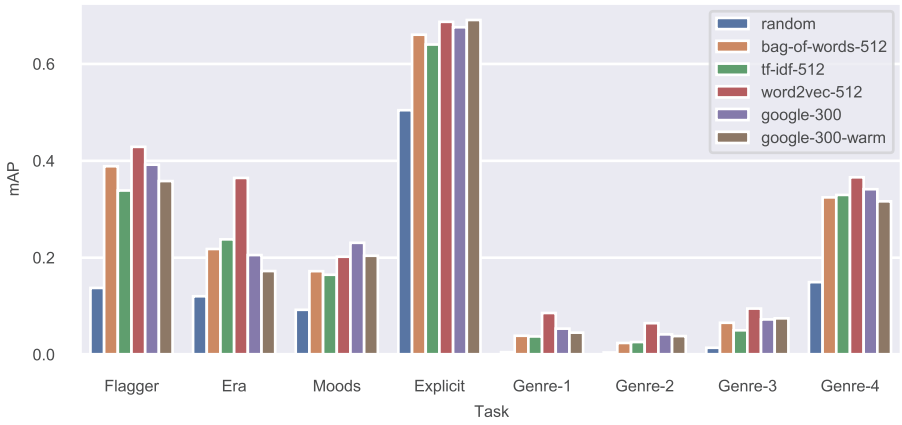


Fig. 2: Word-level embedding experiments, showing mAP on each tagging task. Document embeddings obtained by averaging/summing embeddings across words.

Figure 3, and show that in fact over 80% of the mean average precision can be obtained from 1% of the data (around 170,000 songs).

5 Conclusions

In this paper, we provided a comprehensive quantitative analysis of word2vec style embeddings for music tagging. On a range of challenging tagging tasks at the scale of millions of songs, we discovered that it is hard to surpass the performance of relatively simple models trained on in-house data. Small improvements to averaging embeddings were shown to be possible through sequence modelling, although results were not conclusive. Experiments on sampled data show that increasing training set size beyond O(1M) songs did not significantly improve tagging performance.

In future work, we are interested about the idea of extending our embedding framework to languages beyond English, and also seeing how useful our embeddings are as a source of side information in tasks such as music recommendation.

References

1. Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020.
2. Rémi Delbouys, Romain Hennequin, Francesco Piccoli, Jimena Royo-Letelier, and Manuel Moussallam. Music mood detection based on audio and lyrics with deep neural net. *arXiv preprint arXiv:1809.07276*, 2018.

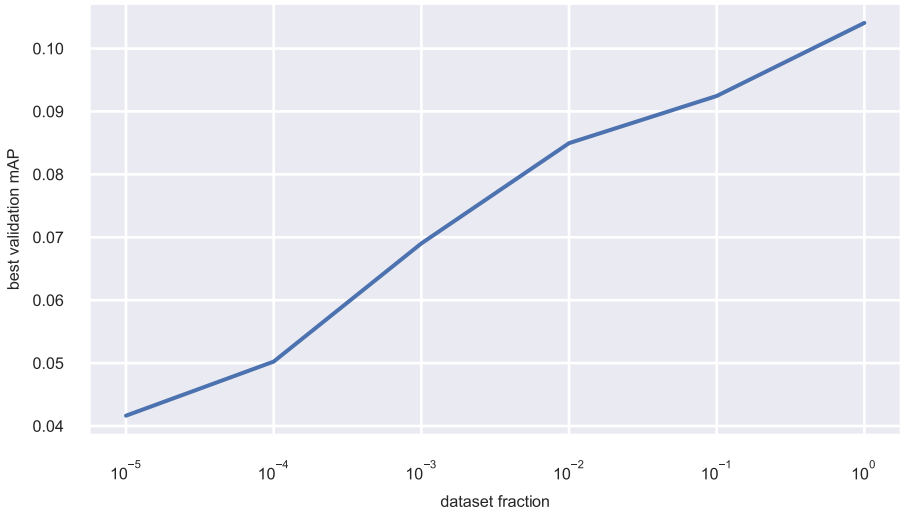


Fig. 3: Effect of the dataset sample size on overall mean Average Precision for the tagging task and `word2vec-512` model.

3. Michael Fell and Caroline Sporleder. Lyrics-based analysis and classification of music. In *Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers*, pages 620–631, 2014.
4. Arthur Flexer and Dominik Schnitzer. Album and artist effects for audio similarity at the scale of the web. In *Proceedings of the 6th Sound and Music Computing Conference (SMC-09*, 2009.
5. Zellig S. Harris. Distributional structure. *WORD*, 10(2-3):146–162, 1954.
6. Xiao Hu and J Stephen Downie. Improving mood classification in music digital libraries by combining lyrics and audio. In *Proceedings of the 10th annual joint conference on Digital libraries*, pages 159–168, 2010.
7. Xiao Hu and J Stephen Downie. When lyrics outperform audio for music mood classification: A feature analysis. In *ISMIR*, pages 619–624, 2010.
8. Xiao Hu, J Stephen Downie, and Andreas F Ehmann. Lyric text mining in music mood classification. *American music*, 183(5,049):2–209, 2009.
9. Li-Ping Jing, Hou-Kuan Huang, and Hong-Bo Shi. Improved feature selection approach tfidf in text mining. In *Proceedings. International Conference on Machine Learning and Cybernetics*, volume 2, pages 944–946. IEEE, 2002.
10. Akshi Kumar, Arjun Rajpal, and Dushyant Rathore. Genre classification using word embeddings and deep learning. In *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pages 2142–2146. IEEE, 2018.
11. Cyril Laurier, Jens Grivolla, and Perfecto Herrera. Multimodal music mood classification using audio and lyrics. In *2008 Seventh International Conference on Machine Learning and Applications*, pages 688–693. IEEE, 2008.
12. Quoc Le and Tomas Mikolov. Distributed representations of sentences and documents. In *International conference on machine learning*, pages 1188–1196. PMLR, 2014.

13. Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Combination of audio and lyrics features for genre classification in digital audio collections. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 159–168, 2008.
14. Rudolf Mayer, Robert Neumayer, and Andreas Rauber. Rhyme and style features for musical genre classification by song lyrics. In *ISMIR*, pages 337–342, 2008.
15. Rudolf Mayer and Andreas Rauber. Musical genre classification by ensembles of audio and lyrics features. In *ISMIR*, pages 675–680, 2011.
16. Cory McKay, John Ashley Burgoyne, Jason Hockman, Jordan BL Smith, Gabriel Vigliensoni, and Ichiro Fujinaga. Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. In *ISMIR*, pages 213–218, 2010.
17. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
18. Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *arXiv preprint arXiv:1310.4546*, 2013.
19. Tom O'Hara, Nico Schüler, Yijuan Lu, and Dan Tamir. Inferring chord sequence meanings via lyrics: Process and evaluation. In *ISMIR*, pages 463–468, 2012.
20. Marco Rospocher. Explicit song lyrics detection with subword-enriched word embeddings. *Expert Systems with Applications*, 163:113749, 2021.
21. Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In *Proceedings of the 2011 European Conference on Machine Learning and Knowledge Discovery in Databases*, ECML PKDD'11, page 145–158, 2011.
22. Alex G Smith, Christopher XS Zee, and Alexandra L Uitdenbogerd. In your eyes: Identifying clichés in song lyrics. In *Proceedings of the Australasian Language Technology Association Workshop 2012*, pages 88–96, 2012.
23. Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *arXiv preprint arXiv:1206.2944*, 2012.
24. Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215*, 2014.
25. Alexandros Tsaptsinos. Lyrics-based music genre classification using a hierarchical attention network. In *ISMIR*, 2017.
26. Menno Van Zaanen and Pieter Kanters. Automatic mood classification using tf* idf based on lyrics. In *ISMIR*, pages 75–80, 2010.
27. Michaela Vystrčilová and Ladislav Peška. Lyrics or audio for music recommendation? In *Proceedings of the 10th International Conference on Web Intelligence, Mining and Semantics*, pages 190–194, 2020.
28. Kento Watanabe and Masataka Goto. A chorus-section detection method for lyrics text. pages 351–359.
29. Kento Watanabe and Masataka Goto. Query-by-blending: A music exploration system blending latent vector representations of lyric word, song audio, and artist. In *ISMIR*, pages 144–151, 2019.
30. Guangxu Xun, Kishlay Jha, Ye Yuan, Yaqing Wang, and Aidong Zhang. Meshprobenet: a self-attentive probe net for mesh indexing. *Bioinformatics*, 35(19):3794–3802, 2019.
31. Dan Yang and Won-Sook Lee. Music emotion identification from lyrics. In *2009 11th IEEE International Symposium on Multimedia*, pages 624–629. IEEE, 2009.
32. Teh Chao Ying, Shyamala Doraisamy, and Lili Nurliyana Abdullah. Genre and mood classification using lyric features. In *2012 International Conference on Information Retrieval & Knowledge Management*, pages 260–263. IEEE, 2012.
33. Lei Zhang, Shuai Wang, and Bing Liu. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4):e1253, 2018.

Oktoechos Classification in Liturgical Music Using Musical Texture Features

Rajeev Rajan *, Amlu Anna Joshy, and Varsha Shiburaj

College of Engineering, Trivandrum, Thiruvananthapuram
APJ Abdul Kalam Technological University, Kerala, India
⋆rajeev@cet.ac.in

Abstract. A distinguishing feature of the music repertoire of the Syrian tradition is the system of classifying melodies into eight tunes, called 'oktoēchos'. It inspired many traditions, such as Greek and Indian liturgical music. In oktoēchos tradition, liturgical hymns are sung in eight modes or eight colours (known as eight 'niram', regionally). In this paper, the automatic oktoēchos genre classification is addressed using musical texture features (MTF), i-vectors and Mel-spectrograms through deep learning strategies. The performance of the proposed approaches is evaluated using a newly created corpus of liturgical music in Malayalam. Long-short term memory (LSTM)-based experiment reports the average classification accuracy of 83.76%, with a significant margin over other frameworks. The experiments demonstrate the potential of LSTM in learning temporal information through MTF in recognizing eight modes in oktoēchos system.

Keywords: liturgy, colour, timbral, deep learning.

1 Introduction

Oktoēchos classification in liturgical music (music using in worship) is addressed using deep learning frameworks in the paper. Music plays a vital role in liturgy because music itself is a language that goes beyond even cultures and races. The vast diversity of forms, styles, and functions in the music used for worship makes it challenging to categorize liturgical music. Musical roles have been distributed in different ways in different rites. Indian orthodox church has imbibed this music system into its liturgy through its relationship with the orthodox church in Syria (Antiochian liturgy). A distinguishing feature of the music repertoire of the Syrian tradition is the system of classifying melodies into eight tunes [15]. This musical tradition is transferred to Indian orthodox liturgical music through centuries with hymns in the Malayalam[1]. Most of the hymns used for various feasts and occasions are musically composed under eight tunes. The system of singing the same text in eight different melodies in an eight-week cycle is referred to as the'oktoēchos' [15].

[1] https://en.wikipedia.org/wiki/Malayalam

1.1 Oktoēchos

Western Syriac music is based on the classical tradition prescribed in 'Bethgazzo'[2]. In oktoēchos tradition, liturgical hymns are sung in eight modes, similar to the Greek liturgy. They are a group of eight adaptable melody types, known as eight 'colours' or 'niram' [27]. None of the Syriac melodies may cover eight notes in an octave. It may often cover three or four or five notes. There is a similarity in Syrian/Indian liturgical (Malankara) hymnal music and rāga[3] of Indian art music. But they cannot be taken in equal level because the rāga classification of Indian art music is incomparable in its scientific systematisation. Each rāga has a particular mode and temperament. Oktoēchos can be compared to rāga in the sense that they are also creating passion or rasa during singing [27]. In Indian art music, a hymn in a rāga can be sung or played in another rāga. The same principle is applied in the oktoēchos system that most of the liturgical hymns can be sung in all the eight tunes.

Oktoēchos is considered as a cyclic system because it is performed in a cycle of eight weeks with two colours in a week. Each colour begins with evening prayer of Sunday. If the first colour is used in the evening, the same is continued for the rest of the day. From Monday evening onwards the fifth colour is used. On Tuesday, it is again switched on to the first colour and so on. The next Sunday begins with the second colour. It is continued in the order 1-5; 2-6; 3-7; 4-8; till to the fourth Sunday and on the fifth Sunday onwards the order becomes 5-1; 6-2; 7-3; 8-4.

1.2 Related Work

Although there has been significant work in music genre classification, the proposed task of liturgical music genre classification is first of its kind. Melodic features [23] and local features [28] have been employed well for genre classification task. Researchers used both generative and discriminative models [12, 24] for music classification. Musical texture features are recently used in meter classification works [22, 21, 19]. Music genre classification is addressed using feature fusion in [20]. A model capable of learning distinctive rhythmic structures of different music genres using unsupervised learning is proposed in [16]. In contrast with the standard approaches, model-based distances between time series can take into account the structure of the songs by modelling the dynamics of the parameter sequence [7]. More recent deep learning approaches process spectrograms for the task of music genre classification [17, 3]. Regarding multimodal approaches found in the literature, most of them combine audio and song lyrics [11] through a fusion framework. The proposed task is similar to music genre classification, but shares the textual content across modes is one of the specific traits of the oktoēchos genre system. The aim of the work is to explore the ability of LSTM to capture the long range dependency in learning temporal patterns.

The rest of the paper is organized as follows; Section 2 describes the proposed system followed by the performance evaluation in Section 3. The analysis of results is given in Section 4. Finally, the paper is concluded in Section 5.

[2] Bethgazzo is a Syriac liturgical book that contains a collection of Syriac chants and melodies.

[3] rāga is the fundamental melodic framework for both Carnatic and Hindusthani traditions

2 System Description

2.1 Feature Extraction

It has already been proven that timbral and rhythmic features are useful in genre classification task [1]. In our experiment, we extracted timbral and rhythmic features as musical texture features. Timbral features, namely Mel-frequency cepstral features (MFCC) and low-level timbral feature-set (T_{LF}), are computed in the front-end. Spectral centroid, spectral roll-off, spectral flux, and spectral entropy [13] are extracted as low-level timbral feature set. Besides, features namely tempo, pulse clarity, event density [10] are computed as rhythmic cues (R_F). Event density represents the number of events per unit time in the music piece. It is a measure that captures how easily "listeners can perceive the underlying rhythmic or metrical pulsation of music" [10]. This feature plays an important role in musical genre recognition, in particular, allowing a finer discrimination between genres that present similar average tempo, but that differ in the degree of emergence of the main pulsation over the rhythmic texture [10]. The distribution of pulse clarity for the corpus is shown in Fig. 1. It can be seen that the pulse clarity distribution for niram 1, niram 2 and niram 3 is different from the rest. Low-level timbral features and rhythmic features are computed using MIRToolbox [4].

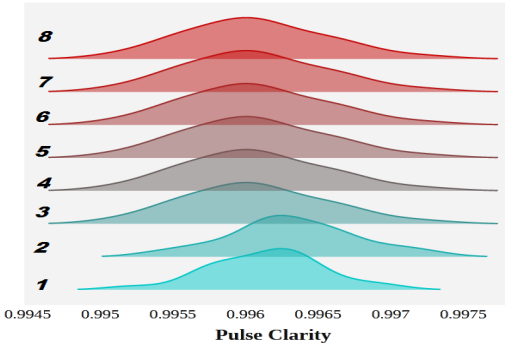


Fig. 1: Distribution of pulse clarity for the colours

Given the success of using i-vectors for speaker and music processing tasks [29, 6], we use the i-vector framework in the proposed task for performance comparison. The i-vector-based statistical feature has been employed well in the task of music genre classification [4]. In i-vector system [5], the high dimensional GMM super vector space (generated from concatenating the mean values of GMM) is mapped to a low dimensional space called total variability space. The target utterance GMM is adapted from a universal background model (UBM) using eigenvoice adaption. The target GMM super vector can be viewed as a shifted version of UBM. Formally, a target GMM super vector M can be written as:

$$M = m + Tw \tag{1}$$

[4] https://www.jyu.fi/hytk/fi/laitokset/mutku/en/research/materials/

where m represents the UBM super vector, T is a low dimensional rectangular total variability (TV) matrix, and w is termed as i-vector. Using training data, the UBM and TV matrix is modeled by expectation maximization. 100 dimensional i-vectors (i_{MFCC}) are computed for each song from MFCC using Alize tool kit [2].

In the final phase, visual representation of audio files, spectrograms are utilized for the proposed task. Since Mel-spectrogram has already been utilized well for music genre classification tasks [25, 8], we also experimented with mel-spectrogram-CNN framework for the proposed task. Mel-spectrogram can be seen as the spectrogram smoothed, with high precision in the low frequencies and low precision in the high frequencies. Mel-spectrogram is computed with frame size of 40 ms and hop size of 10 ms using 128 bins.

2.2 Classification Scheme

We experimented with four classifiers, namely, SVM, DNN, CNN and LSTM. DNN is based on six hidden layered network, which uses 64, 128, 256, 512, 1024, 2048 nodes in successive layers with a dropout of 0.25. The network is trained with the batch size is 32 for 150 epochs by AdaMax optimization algorithm. Relu and softmax have been chosen for hidden and output layers, respectively.

Table 1: LSTM architecture used for the experiment

Sl no.	**Output Size**	**Description**
1	(45,64)	LSTM, 64 hidden units
2	(46, 64)	Dropout (0.25)
3	(1024)	LSTM, 1024 hidden units
4	(1024)	Dropout (0.25)
5	(8)	Dense (8 hidden units)

The proposed CNN has six convolution layers, followed by max-pooling. We use filters with a very small 3×3 receptive fields, with a fixed stride of one and increase the number of filters for the layer by a factor of 2 after every layer. Global max-pooling is adopted in the final max-pooling layer, which is then fed to a fully connected layer. The training is done with 100 epochs by optimizing the categorical cross-entropy between predictions and targets using Adam optimizer, with a learning rate of 0.001.

LSTM architecture shown in Table 1 effectively utilized to track the temporal pattern embedded in the modes of the music. LSTM-RNNs can capture long-range temporal dependencies by overcoming the vanishing gradient problem in conventional RNNs [26]. RNN tap inherent temporal pattern embedded within the frame-wise computed MTF. Deep learning schemes and SVM are implemented using and Keras-TensorFlow and LibSVM, respectively.

Table 2: Overall classification accuracy for the experiments

Sl.No	Feature	Method	Accr.(%)
1	MFCC+ T_{LF} + R_F	SVM	42.65
2	MFCC + T_{LF}+ R_F	DNN	48.70
3	i_{MFCC} + T_{LF} + R_F	DNN	50.00
4	Mel-spectrogram	CNN	52.60
5	**MFCC +** T_{LF}**+** R_F	**LSTM**	**83.76**

3 Performance Evaluation

3.1 Database

A database is created in a studio environment and it consists of eight nirams (colours), with 384 audio tracks with duration 25 to 45 sec per file. No accompaniments were there in the audio files. A total of 15 professional singers in the age group 12 to 50, were participated in the data recording and the whole session was recorded at 44.1kHz. All the singers were very much familiar with the singing modes in 'oktoēchos'. Malayalam hymns were collected from the liturgical book of Indian Orthodox church. The recordings were made niram by niram in successive sessions using a high-quality microphone, A few audio files can be accessed at https://sites.google.com/view/audiosamples-2020/. During experimentation, 60% files of the dataset are used for training, 10% is used for validation and the rest for testing.

3.2 Experimental set-up

MFCCs (39 dim comprising 13 dim MFCC, its delta and delta-delta features), timbral (T_{LF}, 4 dim) rhythmic (R_F, 3 dim) are frame-wise computed with a frame width of 40 ms and hop size of 10 ms and fused in feature-level to obtain 46-dimensional MTF. In the i-vector experimental phase, 100-dimensional i-vectors are computed using 128 mixture GMM from MFCC using Alize tool-kit [2]. UBM model is trained using features derived from the auxiliary database comprising audio file other than the files in the corpus. Auxiliary database, comprising 300 audio files (duration 25-35ms) of liturgical music category, is prepared in a studio environment. The songs from the training data are used for modelling the total variability matrix T by Eigen voice adaption. In the fusion scheme, track level aggregated timbral (T_{LF}) and rhythmic (R_F) features are concatenated with track-level computed i-vectors. Following the evaluation method widely used in the MIR tasks, we computed the precision and recall and the F1 measure as basic evaluation metrics for the performance.

4 Results and Analysis

The results are tabulated in Table 2. As per the table, the average classification accuracy of 42.66%, 48.70%, 50.00%, 52.60% and 83.76% are reported for SVM, DNN, i-vector

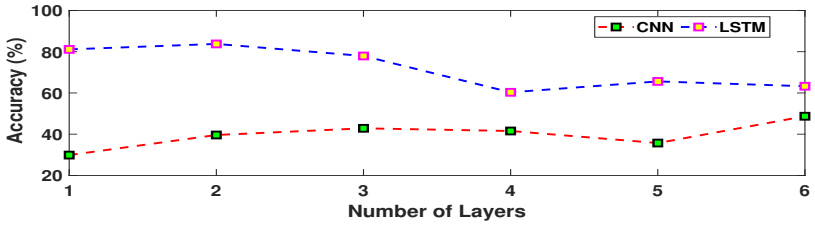


Fig. 2: Accuracy with number of layers for CNN and LSTM

framework, Mel-spectrogram-CNN and LSTM, respectively. It is worth noting that the LSTM outperforms other approaches with a significant margin. It is reasonable to say that time pattern capturing scheme is needed in order to recover more relevant information from temporal embedded musical traits [7]. Experiments show that the LSTM approach is promising for the given task, improving on the case where the dynamics are not taken into account, and a stationary characterization of the sequences is employed. LSTM utilized musical textural features to capture song dynamics effectively to perform oktoēchos classification. It is shown in [4] that the important music elements can be captured by i-vectors and may potentially benefit to the classification of music signal. A possible cause of the low value of accuracy in the given experimental set-up may potentially due to the inability to capture the rhythmic-temporal dynamics well with the given UBM framework. Besides, aggregation of musical texture features to track-level might have deteriorated the performance.

The performance with varying the number of layers of the network is shown in Fig. 2. For the CNN framework, the result improved, as the number of layers increased up to six and then saturated due to overfitting. It is due to the fact that as n increases, the model grows in-depth, and the upper layers find efficient feature representations that are invariant to small perturbations leading to better model generalization. The authors [14] emphasize the need for more training data in the visual representation-based approaches for the genre classification task. It is stated that CNN needs a large size of data to achieve better results since it is not successful enough for less data [9]. An elegant solution to this problem is data augmentation, by which deformations to a collection of annotated training samples results in additional training data. During LSTM approach, maximum accuracy is obtained for two layers as seen in lower-pane in Fig. 2. The proposed experiment validates the claim that temporal information has effectively been learned by MTF-LSTM framework. The experimental insights in [18] show that the performance of the system depends on the temporal architecture, which is basically designed by considering the musical domain knowledge.

The normalized confusion metrics of LSTM is plotted in Fig. 3. Class-wise classification accuracy of all nirams are greater than 70% for LSTM. Niram 5 and niram 7 report accuracy greater than 90%. Class wise accuracy can be examined from the bar plot given in Fig. 4 from all phases. The significant improvement in class-wise accuracy of nirams 1, 3, 7, and 8 over CNN based framework can be seen from the plot. The performance can potentially be improved using data augmentation and proper choice of architecture. The performance metrics precision, recall and F1 score for all the five

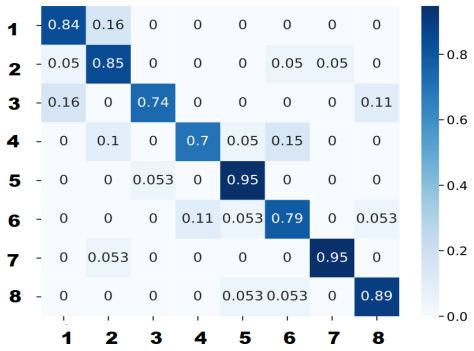


Fig. 3: Normalized Confusion Matrices for MTF-LSTM

approaches are given in Table 3. The average F1 measure of 0.43, 0.50, 0.50, 0.52, 0.84 are reported for SVM, DNN, i-vector-DNN, CNN and LSTM, respectively. The high values of precision, recall and F1 score show the significance of LSTM for the proposed task. Fig. 5 visualizes the output vectors produced by the snippets for the last

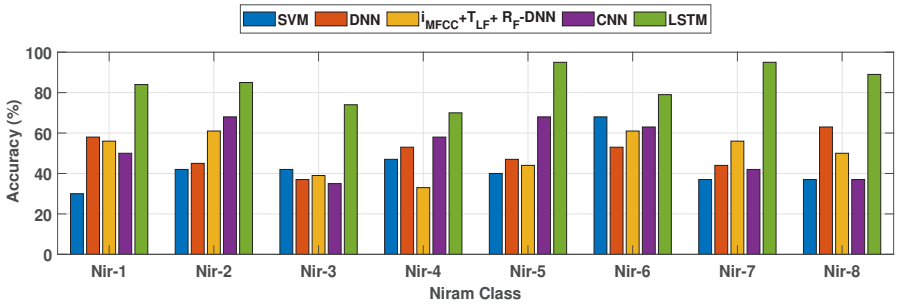


Fig. 4: Class-wise performance for entire phases of the experiments

dense layer of the trained LSTM network using t-SNE. Note that there is good clustering (as represented with colour) and a general separation of different classes for LSTM. It is important to note the effectiveness of LSTM in the proposed task without using any modelling data or augmentation data as that of i-vector or CNN methodologies. Since the results show the promise of temporal pattern learning, other frameworks have to be experimented to investigate the potential of the proposed approach.

5 Conclusion

Oktoēchos classification is addressed in this paper. The performance of the proposed approaches is evaluated using a newly created corpus of Liturgical music in Malayalam.

Table 3: Precision (P), recall (R), and F1 measure

SL.No	**Colour**	**MFCC+T_{LF}+R_F-SVM**			**MFCC+T_{LF}+R_F-DNN**			**$\mathbf{i}_{\mathbf{MFCC}}$+$T_{LF}$+$R_F$-DNN**			**Mel-spectrogram-CNN**			**MFCC+ T_{LF}+R_F- LSTM**		
		P	R	F1	P	R	F1	P	R	F1	P	R	F1	P	R	F1
1	Niram-1	0.55	0.30	0.39	0.35	0.58	0.44	0.48	0.56	0.51	0.42	0.50	0.45	0.80	0.84	**0.82**
2	Niram-2	0.40	0.42	0.41	0.36	0.45	0.40	0.46	0.61	0.52	0.52	0.68	0.59	0.74	0.85	**0.79**
3	Niram-3	0.25	0.42	0.31	0.32	0.37	0.34	0.54	0.39	0.45	0.70	0.35	0.47	0.93	0.74	**0.82**
4	Niram-4	0.64	0.47	0.55	0.71	0.53	0.61	0.46	0.33	0.39	0.69	0.58	0.63	0.88	0.70	**0.78**
5	Niram-5	0.62	0.40	0.48	0.53	0.47	0.50	0.40	0.44	0.42	0.54	0.68	0.60	0.86	0.95	**0.90**
6	Niram-6	0.43	0.68	0.53	0.62	0.53	0.57	0.52	0.61	0.56	0.55	0.63	0.59	0.75	0.79	**0.77**
7	Niram-7	0.33	0.37	0.35	0.50	0.35	0.41	0.59	0.56	0.57	0.47	0.42	0.44	0.95	0.95	**0.95**
8	Niram-8	0.54	0.37	0.44	0.80	0.63	0.71	0.60	0.50	0.55	0.44	0.37	0.40	0.85	0.89	**0.87**
	Macro	0.47	0.43	**0.43**	0.52	0.48	**0.50**	0.50	0.50	**0.50**	0.54	0.53	**0.52**	0.85	0.84	**0.84**

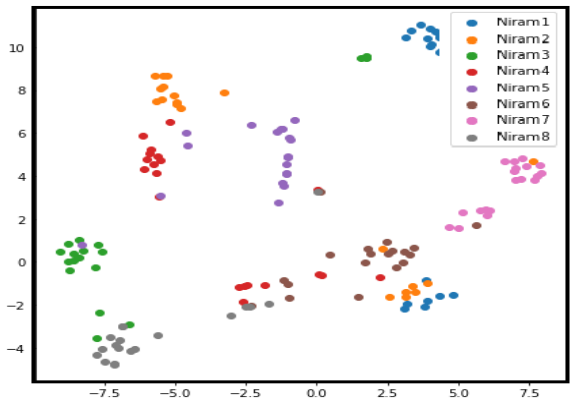


Fig. 5: t_{SNE} plot from LSTM

The evaluation shows the potential of MTF-LSTM framework in Oktoecẖos classification with an average classification accuracy of 83.76%. Since the Greek liturgy and Gregorian chant also share similar musical traits with Syrian tradition, the musicological insights observed can potentially be applied to those traditions as well.

References

1. Baniya, B.K., Ghimire, D., Lee, J.: Automatic music genre classification using timbral texture and rhythmic content features. Proc. of 17th Int. Conference on Advanced Communication Technology pp. 434–443 (2015)
2. Bonastre, J.F., Wils, F., Meignier, S.: Aliźe, a free toolkit for speaker recognition. in Proc. of Interspeech **1**, 737–740 (01 2005)
3. Choi, K., Fazekas, G., Sandler, M., Cho, K.: Convolutional recurrent neural networks for music classification. in Proc. of IEEE Int. Conference on Acoustics, Speech and Signal Processing pp. 2392–2396 (2017)
4. Dai, J., Xue, W., Liu, W.: Multilingual i-vector based statistical modeling for music genre classification. Proc. of Interspeech pp. 459–463 (2017). https://doi.org/10.21437/Interspeech.2017-74
5. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing **19**, 788–798 (2011)
6. Eghbal-zadeh, H., Lehner, B., Schedl, M., Widmer, G.: I-vectors for timbre-based music similarity and music artist classification. in Proc. of 16th Int. Society for Music Information Retrieval Conference pp. 554–560 (2015)
7. Garcia-Garcia, D., Arenas-Garcia, J., Parrado-Hernandez, E., Diaz-de Maria, F.: Music genre classification using the temporal structure of songs. in Proc. of IEEE Int. Workshop on Machine Learning for Signal Processing (2010)
8. Ghosal, D., Kolekar, M.H.: Music genre recognition using deep neural networks and transfer learning. in Proc. of Interspeech pp. 2087–2091 (2018)
9. Kaya, M., Bilge, S.H.: Deep metric learning: A survey. Symmetry **11**(9), 1–26 (2019)

10. Lartillot, O., Eerola, T., Toiviainen, P., Fornari, J.: Multi-feature modeling of pulse clarity: Design, validation and optimization. in Proc. of the 9th Int. Conference on Music Information Retrieval pp. 1–5 (2008)
11. Laurier, C., Grivolla, J., Herrera, P.: Multimodal music mood classification using audio and lyrics. in Proc. of Seventh IEEE Int. Conference on Machine Learning and Applications pp. 688–693 (2008)
12. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. in Proc. of 26th Int. ACM Conference on Research and Development in Information Retrieval pp. 282–289 (2003)
13. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. in Proc. of the 26th Annual Int. ACM Conference on Research and development in information retrieval pp. 282–289 (2003)
14. Liua, C., Fengb, L., Liuc, G., Wangd, H., Liub, S.: Bottom-up broadcast neural network for music genre classification. Pattern Recognition Letters pp. 1–7 (2019)
15. Palackal, J.: Oktoechos of the syrian orthodox churches in south india. Ethnomusicology **48**, 229–250 (2004)
16. Pesek, M., Leonardis, A., Marolt, M.: An analysis of rhythmic patterns with unsupervised learning. Applied Science pp. 1–22 (2020)
17. Pons, J., Lidy, T., Serra, X.: Experimenting with musically motivated convolutional neural network. in Proc. of Int. Workshop on Content-Based Multimedia Indexing pp. 1–5 (2016)
18. Pons, J., Serra, X.: Randomly weighted cnns for (music) audio classification. in Proc. of IEEE Int. Conference on Acoustics, Speech and Signal Processing pp. 336–340 (2019)
19. Rajan, R., Kumar, A.V., Babu, B.P.: Poetic meter classification using i-vector-mtf fusion. In: INTERSPEECH (2020)
20. Rajan, R., Murthy, H.A.: Music genre classification by fusion of modified group delay and melodic features. In: 2017 Twenty-third National Conference on Communications (NCC). pp. 1–6 (2017). https://doi.org/10.1109/NCC.2017.8077056
21. Rajan, R., Raju, A.A.: Poetic meter classification using acoustic cues. In: 2018 International Conference on Signal Processing and Communications (SPCOM). pp. 31–35 (2018). https://doi.org/10.1109/SPCOM.2018.8724426
22. Rajan, R., Raju, A.A.: Deep neural network based poetic meter classification using musical texture feature fusion. In: 2019 27th European Signal Processing Conference (EUSIPCO). pp. 1–5 (2019). https://doi.org/10.23919/EUSIPCO.2019.8902998
23. Salamon, J., Rocha, B., Gomez, E.: Musical genre classification using melody features extracted from polyphonic music signals. in Proc. of IEEE Int. Conference on Audio, Speech, and Signal Processing pp. 81–85 (2012)
24. Shao, X., Xu, C., Kankanhalli., M.S.: Unsupervised classification of music genre using hidden Markov model. in Proc. of IEEE Int. Confernce on Multimedia and Expo, **3**, 2023–2026 (2004)
25. Sukhavasi, M., Adappa, S.: Music theme recognition using CNN and self-attention. preprint arXiv:1911.07041 (2019)
26. Tang, C.P., Chui, K., Yu, Y., Zeng, Z., Wong, K.: Music genre classification using a hierarchical long short term memory model. in Proc. of Int. Conference on Information Retrieval, Japan pp. 521–526 (2018)
27. Vysanethu, P.: Musicality makes the malankara liturgy musical (moran etho 2). St.Ephrem Ecumenical Research Institute, Kottayam, Kerala, India (2004)
28. Wulfing, J., Riedmille, M.: Unsupervised learning of local features for music classifications. in Proc. of Int. Society for Music Information Retrieval Conference. pp. 139–144 (2012)
29. Zhong, J., Hu, W., Soong, F., Meng, H.: DNN i-vector speaker verification with short, text-constrained test utterances. in Proc. of Interspeech pp. 1507–1511 (2017). https://doi.org/10.21437/Interspeech.2017-1036

Modelling Moral Traits with Music Listening Preferences and Demographics

Vjosa Preniqi[1], Kyriaki Kalimeri[2], and Charalampos Saitis[1]

[1] Centre for Digital Music, Queen Mary University of London, London UK
[2] ISI Foundation, Turin, Italy
v.preniqi@qmul.ac.uk

Abstract. Music has always been an integral part of our everyday lives through which we express feelings, emotions, and concepts. Here, we explore the association between music genres, demographics and moral values employing data from an ad-hoc online survey and the Music Learning Histories Dataset. To further characterise the music preferences of the participants the generalist/specialist (GS) score employed. We exploit both classification and regression approaches to assess the predictive power of music preferences for the prediction of demographic attributes as well as the moral values of the participants. Our findings point out that moral values are hard to predict (.62 $AUROC_{avg}$) solely by the music listening behaviours, while if basic sociodemographic information is provided the prediction score rises to 4% on average (.66 $AUROC_{avg}$), with the Purity foundation to be the one that is steadily the one with the highest accuracy scores. Similar results are obtained from the regression analysis. Finally, we provide with insights on the most predictive music behaviours associated with each moral value that can inform a wide range of applications from rehabilitation practices to communication campaign design.

1 Introduction

Music played a fundamental role in the evolution of societies being tightly related to communication, bonding, and cultural identity development [14]. Influencing a wide range of cognitive functions such as reasoning, problem-solving, creativity, and mental flexibility [17], musical taste is also known to be strongly related to personality [7] and political orientation [6]. Musical sophistication is also shown to be related to personality traits regardless of demographics or musicianship level [10].

More recently, scientists aside from the traditional self-reported surveys [6], employed digital data and in particular online music streaming [2] and social media [20] data to assess music preferences. Employing data from the myPersonality Facebook project, Nave et al. [20], found that both people's reactions to unfamiliar music samples and "likes" for music artists predicted personality traits. Krismayer et al. [13] studied the Last.fm platform showing that the music listening behaviours can predict demographics, including age, gender, and nationality. More recently, Anderson et al. [2] presented evidence about the connection between personalities and music listening preferences studying Spotify music streaming data.

Building on comparable interactionist theories, we set to explore the less attended relation between moral values and music preferences. We operationalise morality according to the Moral Foundations Theory (MFT) [9], which defines five moral traits, namely *Care/Harm, Fairness/Cheating, Loyalty/Betrayal, Authority/Subversion*, and *Purity/Degradation*. These can further collapse into two superior moral foundations: of *Individualising*, compounded by fairness and care, that asserts that the basic constructs of society are the individuals and hence focuses on their protection and fair treatment, and of *Binding*, that summarises purity, authority and loyalty, and is based on the respect of leadership and traditions.

Moral values are considered to be higher psychological constructs than the more commonly investigated personality traits yet they have attracted less attention from music scientists. In recent literature there are indications that negative emotions enforced by types of music can worsen moral judgement [3] although that study did not rely on a psychometrically validated theory like the MFT. Kalimeri et al. [12] demonstrated the predictability of moral foundations from a variety of digital data including smartphone usage and web browsing. Their results showed that moral traits and human values are indeed complex, and thus harder to predict compared to demographics, nevertheless, they provide a realistic dimension of the possibilities of modelling moral traits for delivering better targeted and more effective interventions.

Here, we train classification and regression models which infer on self-reported survey data regarding the music preferences. We thoroughly assess the rapresentativity of our data, not only in terms of sociodemographic attributes but also from music behavioural patterns comparing against the open access dataset of music learning histories dataset (MLHD). Our results show that moral values are indeed predictable from music preference information and in line with the findings of the related literature. Further, we discuss the most predictive music behaviours, contributing to an in-depth understanding of the moral profiles. Such insights are fundamental to the broader picture since moral values are a key element in the decision making process on several societal issues [11, **?**]. Modelling moral values from music represents a great opportunity for improving recommendation systems; designing online streaming applications with user well-being in focus [18]; increasing engagement to communication campaigns for social good applications.

2 Data Collection and Feature Engineering

Here, we employ data from a third-party survey administered online for a general scope marketing project. The survey consists of 2,003 participants (51% females) from 12 different regions in Canada. The participants filled in, among other items, information about basic demographic attributes, including age, gender, education, and political views. They also completed the validated Moral Foundations questionnaire [9], while stated their preferences on 13 music genres (on a 5-point Likert scale where 1 = strongly dislike and 5 = strongly like). The considered music genres were: alternative pop/rock, christian, classical, country, folk, heavy metal, rap/hip-hop, jazz, latin, pop, punk, R&B, and rock. These genres were set from the survey creators and were not further described to the respondents. Even so, they are commonly used to define general musical tastes among

Table 1. Summary of the survey dataset (cleaned) with major demographic attributes utilised for this research work.

Attributes	**Demographics**	**Sample size** (N = 1062)
Age	18-24	80 (7.5%)
	25-34	154 (14.5%)
	35-44	205 (19.3%)
	45-54	205 (21.9%)
	55-64	187 (17.6%)
	65+	203 (19.1%)
Gender	Male	474 (44.6%)
	Female	588 (55.3%)
Education	Less than High School	35 (3.2%)
	High school graduate	195 (18.3%)
	Some College	154 (14.5%)
	Trade or professional school	115 (10.8%)
	College Graduate	349 (32.8%)
	Post Graduate work or degree	205 (19.3%))
Political Party	Conservative	328 (30.8%)
	Liberal	279 (26.2%)
	NDI (New Democratic Party)	184 (17.3%)
	Green Party	66 (6.2%)
	Party Quebecois	56 (5.2%)
	I don't vote	149 (14%)

non-musician respondents. To justify these genres and observe if there is any affiliation between survey reported preferences and digital music listening patterns, we explored digital data of 1062 Canadian listeners extracted from the Music Learning Histories Dataset (MLHD) [22] with a similar age and gender distribution to our survey.

Moving on to our survey data, to make sure that participants were paying attention to the survey questions, two "catch questions" were included, which we later used to filter the data. After excluding these users we were left with 1,062 participants (55% females), a sample size substantially higher than previous survey-based studies [7, 6]. Table 1 summarises the demographic features of our dataset.

We then applied a factor analysis using principal axis factoring with promax rotation to identify the major dimensions of participants' music preferences. A 5-factor solution was retained, which explained 67% of total data variance: {jazz, classical, latin}, {punk, heavy metal, rap/hip-hop}, {pop, R&B}, {country, Christian, folk}, and {rock, alternative pop/rock} (genres ordered by decreasing factor loading). These factors are in line with the ones obtained in related studies [7].

To quantify the respondents' diversity in music preferences, we employed an adapted version of the generalist-specialist (GS) score, inspired by the work of Anderson et al. [1]. The projections of the 13 genres onto the five factors were considered as that genre's vector representation in the "preference space". Intuitively, generalists versus specialists

Table 2. Detailed list of the experiments we performed with the list of features employed as predictors in each one of them.

ID	Features Employed as Predictors
EX1	13 Music Genres
EX2	5 factors
EX3	GS score
EX4	13 Music Genres, Age, Gender
EX5	13 Music Genres, Age, Gender, Education
EX6	13 Music Genres, Age, Gender, Education, Political Views

will have genre vectors spread apart versus close together in the preference space. We calculate the user centroid $\overrightarrow{ct_i}$ of genre vectors representing the loadings of genres on the 5 factors $\overrightarrow{l_j}$, weighted by the number of genre scores rated by each respondent w_j. The GS score is the cosine similarity between a genre vector and the preference-weighted average of a users' genre vectors:

$$GS(u_i) = \frac{1}{\sum w_j} \cdot \sum w_j \frac{\overrightarrow{l_j} \cdot \overrightarrow{ct_i}}{\| \overrightarrow{l_j} \| \cdot \| \overrightarrow{ct_i} \|} \quad , \qquad \overrightarrow{ct_i} = \frac{1}{\sum w_j} \cdot \sum w_j \overrightarrow{l_j}$$

3 Experiments and Results

Exploratory Analysis. As a first step we assess the correlation between musical genres' preferences, demographics, political views and moral traits. We observed a positive Spearman correlation of age with Christian music, classical, country and folk music genres ($\rho_s = \{0.18, 0.21, 0.20, 0.25\}$), while heavy metal, hip-hop/rap and punk were more preferred by younger ages, whereas older people expressed their dislike towards these genres ($\rho_s = \{$-0.22, -0.38, -0.38 $\}$). Education was positively related with classical music, jazz and latin music ($\rho_s = \{0.22, 0.13, 0.13\}$), indicating that people with higher education preferred these genres. Loyalty, authority and purity were positively correlated with Christian music ($\rho_s = \{0.18, 26, 38\}$) and country music ($\rho_s = \{0.17, 20, 21\}$). Looking at the political views of the respondents, conservatives were positively correlated with Christian genre and country ($\rho_s = \{0.12, 0.12\}$) and negatively correlated to hip-hop/rap and punk ($\rho_s = \{$-0.17, $-0.15\}$).

Further, we assessed whether the obtained self-reported responses of the questionnaire are in line with digital music listening data. From the MLHD dataset [22] we extracted artists' genres using MusicBrainz identifiers. From the survey data we discerned that the top 10 most preferred genres were: rock, pop, alternative pop-rock, classical, r&b, country, jazz, folk, latin, and hip-hop/rap. Similar trends were encountered in the music listening histories of Canadian users in MLHD where the 10 most frequently listened genres were: rock, alternative rock, pop-rock, pop, electronic, folk, punk, jazz, heavy metal, and hip-hop.

Moral Values Classification. Our main research question is whether we can predict peoples' moral values from their music preferences. To answer this question, we postulate

Table 3. Moral traits classification with XGBoost, average weighted AUROC and standard deviation over 5-fold cross-validation (baseline is .50).

	EX1	EX2	EX3
Care	**.57 (3.7)**	.54 (2.1)	.52 (1.5)
Fairness	**.56 (2.9)**	.52 (1.1)	.48 (2.7)
Authority	**.63 (0.8)**	.60 (1.1)	.49 (1.7)
Purity	**.69 (2.8)**	.65 (3.0)	.57 (2.3)
Loyalty	**.61 (2.4)**	.56 (1.9)	.48 (3.1)
Individ.	**.55 (3.5)**	.51 (0.8)	.50 (1.6)
Binding	**.67 (2.4)**	.63 (2.2)	.52 (1.9)

Table 4. Moral traits classification with XGBoost for different predictors (see Table 2 where they are defined). Models evaluated based on AUROC and standard deviation over 5-fold cross-validation (baseline is 50).

	EX1	EX4	EX5	EX6
Care	.57 (3.7)	.62 (3.2)	.62 (3.0)	**.63 (2.3)**
Fairness	.56 (2.9)	.58 (2.5)	.57 (2.3)	**.62 (4.3)**
Authority	.63 (0.8)	.64 (1.6)	.65 (2.0)	**.66 (1.6)**
Purity	.69 (2.8)	.71 (3.0)	**.71 (1.4)**	.71 (1.6)
Loyalty	.61 (2.4)	**.67 (3.5)**	.66 (2.2)	.66 (2.9)
Individ.	.55 (3.5)	.59 (2.4)	.59 (3.3)	**.61 (1.8)**
Binding	.67 (2.4)	.71 (3.2)	.70 (2.2)	**.72 (2.9)**

the task as a supervised classification one, developing a series of experiments to assess the predictive power of different variables (see Table 2). We assign the class label "high" to individuals with moral scores higher than the population median for the specific foundation, and "low", otherwise. We perform 5-fold cross-validation on shuffled data (to avoid dependencies in successive data points), with 70% of training and 30% testing data. We opt for the gradient boosting algorithm XGBoost (XGB) as it performed better than Random Forest (RF) and Support Vector Machine (SVM) in this task.

To take into account the effect of unbalanced class labels in the performance metric, we evaluate our models with the area under the receiver operating characteristic (AUROC) metric which is a performance measure for binary classifiers that employs a discrimination threshold to differentiate between a high and a low class [12]. The best model is then chosen as the one that maximized the weighted area under receiver operating characteristic (AUROC) statistic.

Initially, we compared the predictive power of the genre information against the features engineered by us (EX1, EX2, and EX3). We trained one model per moral foundation, and we present the cross validated results in Table 3. We notice that the information obtained directly about the music preferences (EX1) outperforms the features we developed. When comparing the scenarios, we observe that the 5 factors, and the GS score accounting only for part of the variance in the data, did not manage to outperform the explicit information on music preferences. A question that emerges naturally, is

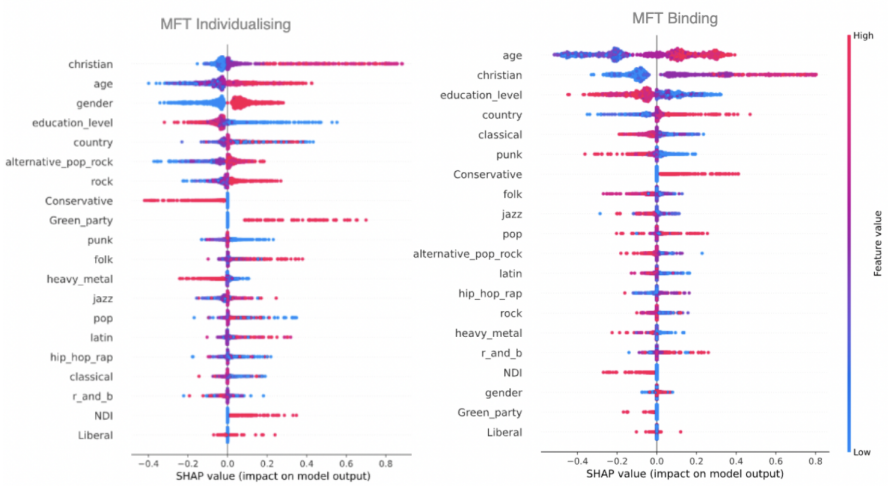


Fig. 1. Feature contributions (via SHAP values). The higher the SHAP value, the more the feature contributes to the moral prediction.

.

whether including knowledge regarding the participants' basic demographic features (i.e. age, political views, education level) will improve the prediction of their moral values. Table 4 summarises the results when age, gender, education and political views are incorporated in the design. As expected, the more information we have about the participants the more precise our predictions become, however, the improvement is minimum. This shows us the importance of music behaviours alone in explaining the variability of our moral values.

Further, we employed SHAP (SHapley Additive exPlanations), a game theory approach developed to explain the contribution of each feature to the final output of any machine learning model [15]. SHAP values provide both global and local interpretability, meaning that we can assess both how much each predictor and each observation, respectively, contribute to the performance of the classifier. SHAP's output helps to understand the general behaviour of our model by assessing the impact of each input feature in the final decision, thus enhancing the usefulness of our framework (Figure 1).

Moral Values Regression. Data binning is a common way to aggregate information and facilitate the classification tasks. However, there are known issues to dichotomisation of variables which often lead to misleading results [16]. Here, to ensure that the most predictive features as emerged from the classification process are indeed descriptive of the respective moral trait, we conducted a regression analysis. At this point, the aim is to understand whether we can estimate the original moral scores (predicting the quantity) based on our explanatory variables in disposition (i.e., music genres ad demographics).

To do so, we trained an XGBoost Regressor for each moral foundation. We maintained the same experimental designs and settings as in the classification task. For

Table 5. Mean Absolute Error (MAE) and standard deviation over 5-fold cross-validation for XGBoost regression on music preference features (see Table 2).

	EX1	EX2	EX3
Care	3.86 (13.2)	**3.72 (10.9)**	3.89 (7.0)
Fairness	**3.27 (11.1)**	3.28 (9.6)	3.55 (8.7)
Authority	**4.19 (23.3)**	4.20 (16.7)	4.47 (13.9)
Purity	**4.86 (19.7)**	4.99 (25.0)	5.35 (21.0)
Loyalty	4.46 (12.1)	**4.33 (19.4)**	4.64 (11.6)
Individ.	3.23 (9.5)	**3.17 (8.5)**	3.35 (9.9)
Binding	3.86 (15.1)	**3.79 (6.3)**	4.22 (18.5)

Table 6. Mean Absolute Error (MAE) and standard deviation over 5-fold cross-validation for XGBoost regression on music preference and demographic features (see Table 2).

	EX1	EX4	EX5	EX6
Care	3.86 (13.2)	3.72 (6.2)	3.71 (9.3)	**3.60 (8.0)**
Fairness	3.27 (11.1)	3.25 (8.2)	3.19 (10.5)	**3.12 (13.4)**
Authority	4.19 (23.3)	4.14 (15.9)	4.10 (9.3)	**4.09 (11.0)**
Purity	4.86 (19.7)	4.86 (20.4)	4.74 (18.9)	**4.71 (15.8)**
Loyalty	4.46 (12.1)	**4.19 (22.8)**	4.20 (18.7)	4.21 (18.7)
Individ.	3.23 (9.5)	3.17 (14.4)	3.17 (8.0)	**3.0 (8.9)**
Binding	3.86 (15.1)	3.80 (11.0)	3.76 (13.1)	**3.74 (5.4)**

evaluation, we used Mean Absolute Error (MAE). These options allow for a direct comparison of the most predictive features with the ones emerged from the classification task (Table 5). We noticed that as in the classification task, when adding information to the models the MAE decreases indicating that the model fits the data better. Also in this case the gain of adding more information is relatively small with respect to the music genres alone.

We visualised the most predictive features using again the Shap values (see Figure 2). Interestingly, the christian music genre appears again as the most important predictor for both the Binding and Individualising traits. The feature importance for the output of the XGboost regressor, is in line with the feature significance obtained with the classification approach. The same holds for all the moral foundations which are not depicted here for spacing issues.

4 Discussions and Conclusions

Henry Wadsworth Longfellow wrote, "Music is the universal language of mankind." Contemporary research has found converging evidence that people listen to music that reflects their psychological traits and needs and help express emotions, cultures, values and personalities. In this paper, we analysed the less explored links between musical preferences, demographics (age, gender, political views, and education level) and Moral

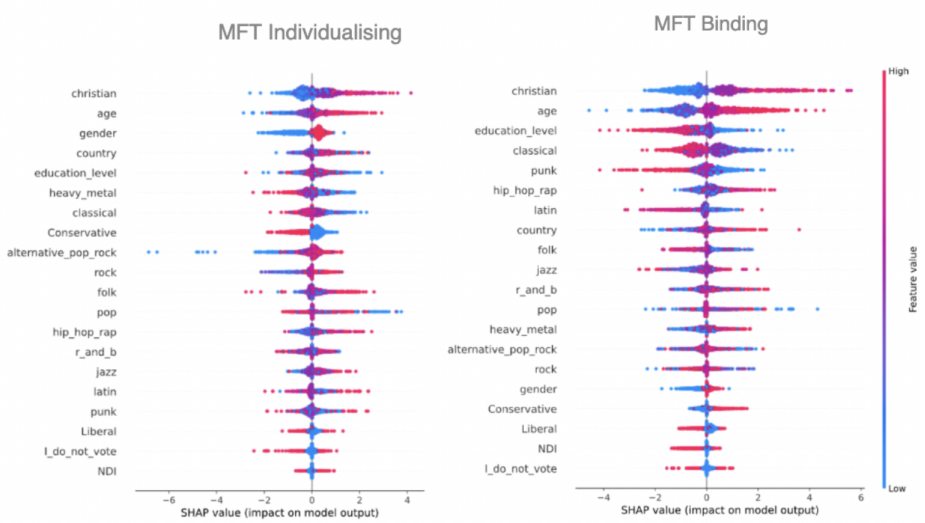


Fig. 2. Features with the most impact on the XGBoost regression model output.

Foundations (MFT [9]). We applied both classification and regression models for moral traits prediction. From classification results, it was inferred that MFT Binding was best predicted with AUROC score 72%, whereas MFT individualising showed weaker results with AUROC score 61%. While, for the regression task the lowest MAE was 3.0 for the Individualising and 3.74 for Social Binding. In both approaches, the most impactful features on inferring morality were christian music and age.

Moral foundations are strongly tied to political views; despite that, the musical features are more predictive than political leanings. Social binding is related to conservative political views [8] - and in fact is predicted by christian, and country music. We notice that people naturally express their moral values through the music they listen to. We instinctively *categorize* objects, symbols, but also people, creating a notion of *social identity*. According to the social identity theory members of a group will seek to find negative aspects to other groups thus enhancing their self-image [21]. Such reasoning reflects on a broad range of attitudes related to stereotype formations [19] but also as we notice here to musical preferences. For instance, people higher in social binding foundations tend to listen to country music which often expresses notions of patriotism. Christian music is also a predictor of this superior foundation, which again fosters the notion of belonging to a group. Across all experiments, Christian music emerged as the most predictive genre. On the other hand, genres such as punk, and hip hop are known to challenge the traditional values and the status quo, hence are preferred by people who strongly value these aspects. Our findings suggest that musical preferences are quite informative of deeper psychological attributes; still there is space for improvement. For instance, we noticed that the care, fairness, and loyalty foundations are harder to predict. To this end we aim to explore musical content analysis, for instance, incorporating

linguistic cues, and the moral valence scores as proposed by Araque et al. [4, 5] on lyrics to further improve the performance.

In future work we aim to delve deeper into the relation between music and morality, and between music and other universal human values, by using passively collected digital traits of music listening behaviours outside a laboratory setting and over a period of time [2], while using self-reported surveys as a solid groundtruth. We will further investigate the association between music listening preferences other psychological aspects such as human values and emotions. Developing data-informed models will help unlock the potential of personalised, uniquely tailored digital music experiences and communication strategies [12, 1]. Predicting the moral values from listening behaviours can provide noninvasive insights on the values or other psychological aspects of populations at a large scale.

5 Acknowledgements

This work was supported by the QMUL Centre for Doctoral Training in Data-informed Audience-centric Media Engineering (2021–2025) as part of a PhD studentship awarded to VP. KK acknowledges support from the "Lagrange Project" of the ISI Foundation, funded by the CRT Foundation. We would like to thank Dr. Robert Raleigh for providing the survey data, and the two anonymous reviewers for their thoughtful comments.

References

1. Anderson, A., Maystre, L., Anderson, I., Mehrotra, R., Lalmas, M.: Algorithmic effects on the diversity of consumption on spotify. In: Proc. Web Conf. 2020. pp. 2155–2165 (2020)
2. Anderson, I., Gil, S., Gibson, C., Wolf, S., Shapiro, W., Semerci, O., Greenberg, D.M.: "just the way you are": Linking music listening on spotify and personality. Soc. Psychol. Personal. Sci. **12**(4), 561–572 (2021)
3. Ansani, A., D'Errico, F., Poggi, I.: 'you will be judged by the music i hear': A study on the influence of music on moral judgement. In: Web Intel. vol. 17, pp. 53–62 (2019)
4. Araque, O., Gatti, L., Kalimeri, K.: Moralstrength: Exploiting a moral lexicon and embedding similarity for moral foundations prediction. Knowl. Based Syst. **191**, 105184 (2020)
5. Araque, O., Gatti, L., Kalimeri, K.: The language of liberty: A preliminary study. In: Companion Proc. Web Conf. 2021. pp. 623–626 (2021)
6. Devenport, S.P., North, A.C.: Predicting musical taste: Relationships with personality aspects and political orientation. Psychol. Music **47**(6), 834–847 (2019)
7. Gardikiotis, A., Baltzis, A.: 'rock music for myself and justice to the world!': Musical identity, values, and music preferences. Psychol. Music **40**(2), 143–163 (2012)
8. Graham, J., Haidt, J., Nosek, B.A.: Liberals and conservatives rely on different sets of moral foundations. J. Personal. Soc. Psychol. **96**(5), 1029–1044 (2009)
9. Graham, J., Nosek, B., Haidt, J., Iyer, R., Koleva, S., Ditto, P.H.: Mapping the moral domain. J. Personal. Soc. Psychol. **101**(2), 366–385 (2011)
10. Greenberg, D.M., Müllensiefen, D., Lamb, M.E., Rentfrow, P.J.: Personality predicts musical sophistication. J. Res. Pers. **58**, 154–158 (2015)
11. Haidt, J., Joseph, C.: Intuitive ethics: How innately prepared intuitions generate culturally variable virtues. Daedalus **133**(4), 55–66 (2004)

12. Kalimeri, K., Beiró, M.G., Delfino, M., Raleigh, R., Cattuto, C.: Predicting demographics, moral foundations, and human values from digital behaviours. Comput. Hum. Behav. **92**, 428–445 (2019)
13. Krismayer, T., Schedl, M., Knees, P., Rabiser, R.: Predicting user demographics from music listening information. Multimed. Tools. Appl. **78**(3), 2897–2920 (2019)
14. Loersch, C., Arbuckle, N.L.: Unraveling the mystery of music: Music as an evolved group process. J. Personal. Soc. Psychol. **105**(5), 777–798 (2013)
15. Lundberg, S.M., Lee, S.I.: A unified approach to interpreting model predictions. In: Proc. 31st Int. Conf. Neural. Inf. Process. Syst. pp. 4768–4777 (2017)
16. MacCallum, R.C., Zhang, S., Preacher, K.J., Rucker, D.D.: On the practice of dichotomization of quantitative variables. Psychol. Methods **7**(1), 19–40 (2002)
17. MacDonald, R., Kreutz, G., Mitchell, L.: Music, health, and wellbeing. Oxford University Press (2013)
18. Mejova, Y., Kalimeri, K.: Effect of values and technology use on exercise: implications for personalized behavior change interventions. In: Proc. ACM Conf. User Model. Adapt. Personaliz. (UMAP) 2019. pp. 36–45 (2019)
19. Miller, S.L., Maner, J.K., Becker, D.V.: Self-protective biases in group categorization: Threat cues shape the psychological boundary between "us" and "them". J. Personal. Soc. Psychol. **99**(1), 62–77 (2010)
20. Nave, G., Minxha, J., Greenberg, D.M., Kosinski, M., Stillwell, D., Rentfrow, J.: Musical preferences predict personality: evidence from active listening and facebook likes. Psychol. Sci. **29**(7), 1145–1158 (2018)
21. Tajfel, H., Turner, J.C., Austin, W.G., Worchel, S.: An integrative theory of intergroup conflict. Org. Id.: A reader **56**(65), 33–47 (1979)
22. Vigliensoni, G., Fujinaga, I.: The music listening histories dataset. In: Proc. Int. Soc. Music Inf. Retr. Conf. (ISMIR). pp. 96–102 (2017)

Classification of 1950 to 1960 Electronic Music Using the VGGish Neural Network and Random Forest

Maurício do V. M. da Costa[1], Florian Zwißler[1], Philip Schwarzbauer[1] and Michael Oehler[1]

[1] Music Technology & Digital Musicology Lab (MTDML), Institute for Musicology and Music Pedagogy, Osnabrück University, Germany
michael.oehler@uos.de

Abstract. This paper presents an approach to extend an ontological database concept aimed at the systematization of Electronic Music. Machine Learning techniques are used to test the significance of empirical investigations on the "output layer" of the production process, namely finished compositions of Electronic Music. As an example, pieces from the era of 1950 to 1960 are being examined, representing the aesthetics of *Musique Concrète* from Paris and *Elektronische Musik* from Cologne. The experiments performed using state-of-the-art techniques suggest the confirmation of measurable differences in the musical pieces from different studios for electronic music that were motivated by aesthetically divergent approaches.

Keywords: electronic music, musique concrète, Elektronische Musik, VGGish, random forest

1 Introduction

1.1 Analysis and systematization of Electronic Music

Despite Electronic Music having existed for many decades, it is still lacking tools to reliably systematize it, the most striking being a shortage of a clear terminology capable of describing the phenomena themselves as well as the processes used to produce them. In most cases, analogies to the strong and established terminologies of instrumental music and sound production [1-5] are being taken as a solution to this problem, not facing the problem that electronic sound production implies a fundamentally different potential that needs to be addressed [6]. This issue is continued in the field of music analysis: only a few attempts have been made to present universally valid tools that allow musicologists to get significant insights into the structure of a piece of Electronic Music. The most valuable source of information at hand is represented by [7, 8] and

the recently revised EMDoku[1], a huge database of Electronic Music that gives insights into all the results of composing with electronically produced sound. A systematization that comprises the conditions of the production of these results is yet to be found. Recently, this topic has received more attention, for example, with regard to *Musique Concrète* [9].

The PRESET research project, which was presented at CMMR 2019, has set out to do basic work to make progress in this direction: a database is being put together collating information from an in-depth survey of several studios for Electronic Music. Exploring their informational resources and bringing them together will open new lines of insight into the nature and relations of the processes involved. To address this issue, it was decided to use a semantic web database with an underlying ontology as a structural and terminological foundation [10]. In connection with the methods of actor-network theory [11] and theories from the field of information systems [12, 13], the working processes within the single studios as well as the connection in between them will display a new perspective on the field.

1.2 Electronic Music in the 1950s: *Musique Concrète* and *Elektronische Musik*

The early period of electronic music was characterized by a vivid debate between two quite different approaches of composing music within the context of an electronic studio. The *Musique Concrète,* which originated from Paris with its founder Pierre Schaeffer and since 1958 organised in the *Groupe de Recherches Musicales* (GRM), and the approach called *Elektronische Musik* (electronic music), which was pursued at the West German Radio in Cologne, most prominently represented by its then leader Herbert Eimert and Karlheinz Stockhausen. The *Musique Concrète* originally set out their experiments from recorded sound, thus integrating the production medium (records and, later on, magnetic tape) within the very first steps of working on sound. The repertoire of sound to create a piece was gained by very simple means of manipulation such as cutting the tape, reversing it, changing its speed, and building loops to generate rhythmic structures. This results in an empirical approach on dealing with sound as a medium to work on, also leading to an elaborate theoretical concept of the nature of sounds that Schaeffer formulated in his *Traité des objets musicaux* [14]. In Cologne, on the other hand, the idea was rather to construct the sound following a pre-structured concept devised by the composer. This strategy, in turn, was strongly connected to the concept of serial music, which favored a view on composition as a formal organization of sets of parameters [15]. It is evident that this view found a perfect fit in the new possibilities of sound creation and organization in an electronic studio.

These two approaches, of course, did not exist separately from one another, and there was a vital interest in each other's musical results. The opposing views on concepts of composition have been broadly discussed [16-18] and have led to the view that there was a remarkable aesthetic difference in these approaches.

Apart from the discussion to what extent this holds true, we decided to take the diverging concepts to an empirical test with the use of Machine Learning techniques.

[1] www.emdoku.de

2 Method

The methods presented in Section 1.1 basically represent a top-down model of systematization. Connecting our efforts to the existing potentials of databases such as EMDoku, we decided to add a bottom-up method of information retrieval in analyzing datasets representing the actual "output" of the studios in Cologne and Paris within a time window ranging from 1950 to 1960 – a period where the aesthetically divergent approaches were most prominent [18]. In doing so, we will try to test methods of empirical analysis and check them for their significance

The experiment consists in using a pre-trained Deep Neural Network (DNN) to convert the audio samples into semantically meaningful embeddings and then training a classifier to learn to identify material from both classes (GRM and WDR) using such high-level embeddings as input features. This way, we propose to empirically assess the existence of differences between recordings of such groups in purely acoustic features. Although this approach does not indicate what those differences are, we intend to pursue an indirect demonstration of their existence, for the only information provided for classification is related to audio content.

The VGGish [19] model was used for the computation of the embeddings. This network is based on the VGG [20] model, which is one of the most used DNN architectures for image recognition, and produces embeddings of 128 samples. In order to prepare the audio data to be processed by this network, first, the audio input signal is collapsed to mono and band-limited to 8 kHz. Then, its spectrogram is computed using the short-time Fourier transform, with a Hann analysis window of 25 ms and a hop of 10 ms. After that, a mel-spectrogram with 64 frequency bands (125 - 7500 Hz) is obtained by remapping the spectrogram time-frequency bins. This mel-spectrogram is then framed into non-overlapping examples of 0.985 s, each example covering the 64 mel bands and 96 time frames of 10 ms each. Finally, this process produces the embeddings for all audio files available by computing the network's outputs and stores them in text files with the same names as their audio counterparts.

Then, the random forest algorithm was used to classify the embeddings produced. To avoid having excerpts of the same musical piece both in the training and test sets by treating the embeddings as independent samples, all the embeddings of each piece were either assigned to the training set or to the test set. For this purpose, a random selection of the pieces was performed with a probability of 70% of each piece being selected as training data and 30% as test data. Since their variability in length is large (ranging from less than a minute to several minutes), considerable differences occur in the actual train/test proportion. This same classification experiment was repeated 10 times and both the average and the standard deviation of the results were computed to illustrate the classification performance. We used the implementation present in the "Scikit learn"[2] framework for the random forest algorithm, set to train an ensemble of 400 decision trees and using its default settings. Smaller number of trees were tested and provided slightly lower performance. Nevertheless, yielding high classification performance and providing a detailed analysis regarding the classification problem itself are not the objective of this paper.

[2] scikit-learn.org

In order to assess the performance, the majority vote of the embeddings within each musical piece was taken to assign the piece's classification. This way, each piece accounted for one sample, instead of their group of embeddings, i.e. pieces from which more than 50% of the embeddings were correctly estimated are considered to be one correctly estimated sample, despite its duration.

The actual lists of pieces used for the analysis were determined through the following: as a first step, all output from both studios within the chosen time interval was identified following the data resources provided by EMDoku, which represents the most reliable resource available. After that, only works that purely consist of electronically produced sounds were selected, thereby excluding all pieces that use sound resources from outside the production processes in discussion. It was also decided to exclude all sorts of functional compositions (e.g. music for radio plays) within those lists to again ensure the validity of the data as examples of the two aesthetic directions. The next step was to retrieve the actual audio material of the pieces. From the list of pieces from the studio of the WDR, it was possible to obtain about 75% of the pieces in question (57 files), making up a total duration of 3.5h. The examples available from the GRM made up a fairly larger amount, with 94 files, totaling roughly 6h of audio material.

It should be noted that we only compared the audio content of these pieces with no regard to spatialization, so from all the pieces, also those that exist in multichannel versions, only mono-mixdown versions were used, due to the characteristics of the architecture adopted for the classification task.

3 Results

The results obtained from this procedure are summarized in Table 1, which shows the average and standard deviation for accuracy, precision, recall and F-measure. Despite the small dataset available, the results suggest that the classifier was capable of identifying differences in the acoustic features related to each aesthetic approach.

Table 1. Overall results.

Measure	Average	Std.
Accuracy	0.82	0.08
Precision	0.89	0.08
Recall	0.66	0.20
F-measure	0.74	0.14

A histogram that represents classification accuracy of the embeddings within each musical piece, i.e. the proportion of correct votes for each class within each piece, is illustrated in Figure 1. As can be observed, the distributions obtained have different characteristics: the classifier was more successful in identifying excerpts from GRM, with voting proportion more concentrated towards 100% than from WDR, which had more diluted classification of the embeddings. In total, the GRM pieces were classified as 82% GRM and 18% WDR, whereas the WDR pieces were estimated to be 47% WDR and 53% GRM.

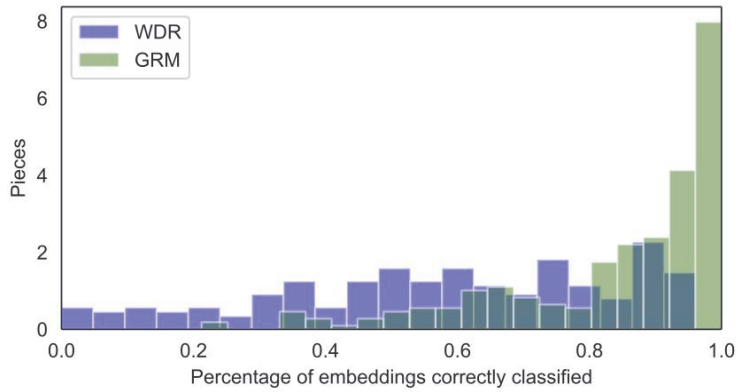


Fig. 1: Histogram of classification accuracy within each musical piece.

The distributions obtained suggest that the classes may have a significant overlap, as expected, but the classification system tended to have a bias towards the GRM class despite all data imbalancement compensation techniques. This may indicate that the GRM pieces might have less variation within the acoustic features of interest for the classification system, whereas the WDR pieces may show a wider variety in such dimensions. Besides, the pieces have a considerable amount of excerpts where the audio material present may severely interfere in this analysis, like background noise or long reverb tails. Nevertheless, the results are informative and serve the purpose of empirically assessing the differences present in sound.

4 Conclusion

The experiments presented in this paper served the initial goal to widen the focus of a database still under construction that aims at facilitating a valid and significant systematization of Electronic Music. The Machine Learning techniques employed to analyze the two specific sets of compositional results of studio work have displayed a specific difference within these sets. A possible consequence of this outcome in interaction with a future ontological database could be to check the technical equipment used within the specific time interval for correspondences and differences, as well as to investigate possible interdependencies of personnel involved. The inclusion of this “bottom up”- method is therefore likely to provide valuable insights and to bring up crucial questions to constantly improve the structure of the database as a whole.

The experimental setup was comprised of two different Machine Learning techniques: a pre-trained deep neural network (VGGish), which uses as input mel-spectrograms of the audio signal and outputs a sequence of high-level embeddings, followed by a random forest classifier, which was trained to differentiate embeddings from both classes under analysis. The musical pieces were then classified using the criterion of majority vote of the classes estimated for their embeddings. The train and test sets were randomly generated from piece selection and the experiment was performed 10 times. No embeddings from the same musical piece were used for both training and testing.

Although the results are not particularly outstanding for a music genre classification task (see Table 1), they do show that there are indeed noticeable differences in the acoustic features extracted from the pieces in these groups. This provides empirical evidence for what was only discussed theoretically in earlier studies.

It is worth mentioning that the VGGish network does not encompass long-term temporal interdependencies of acoustic events, which are a fundamental part of music structure and may reveal hidden patterns that could improve this intricate classification task. For this purpose, we intend to expand this experiment in future work tackling this specific problem by considering the whole sequence of embeddings using a different downstream model, instead of purely classifying each one independently, or even using a deep neural network that takes into account the temporal dimension.

References

1. Hornbostel, E. M. von, Sachs, C.: Systematik der Musikinstrumente. Ein Versuch. Zeitschrift für Ethnologie, 46: pp. 553–590 (1914).
2. Kartomi, M. J.: On Concepts and Classifications of Musical Instruments. The University of Chicago Press, Chicago (1990).
3. Simon, P.: Die Hornbostel/Sachs'sche Systematik der Musikinstrumente: Merkmalarten und Merkmale. Eine Analyse mit zwei Felderdiagrammen. Verlag Peter Simon, (2004).
4. MIMO – Musical Instrument Museums Online, *http://www.mimo-db.eu*, retr. 21/02/28.
5. Montagu, J.: Origins and Development of Musical Instruments. Scarecrow Press, (2007).
6. Kolozali, S., Barthet, M., Fazekas, G., Sandler, M. B.: Knowledge Representation Issues in Musical Instrument Ontology Design. In ISMIR: pp. 465-470 (2011).
7. Davies, H. (Ed.): Répertoire International des Musiques Electroacoustiques: International Electronic Music Catalog. Groupe de Recherches Musicales de l'ORTF (1967).
8. Hein, F., Seelig, T.: Internationale Dokumentation Elektroakustischer Musik. Pfau (1996).
9. Godøy, R. I.: Perceiving sound objects in the musique concrète. In: Frontiers in Psychology, 12, 1702 (2021).
10. Abdallah, S., Raimond, Y., Sandler, M.: An ontology-based approach to information management for music analysis systems. In: AES Convention 120 (2006).
11. Latour, B.: Reassembling the Social: An Introduction to Actor-Network-Theory. Oxford University Press, Oxford (2005).
12. Goldkuhl, G.: Design Theories in Information Systems-a Need for Multi-Grounding. In: Journal of Information Technology Theory and Application (JITTA) 6(2): 7 (2004).
13. Gregor, S.: A Theory of Theories in Information Systems. In: Information Systems Foundations: Building the Theoretical Base: 1-20 (2002).
14. Schaeffer, P.: Traité des Objets Musicaux, Essai Interdisciplines. Le Seuil, Paris (1966)
15. Morawska-Buengeler, M.: Schwingende Elektronen. Tonger, Cologne (1988).
16. v. Blumröder, C.: Die elektroakustische Musik: Eine kompositorische Revolution und ihre Folgen. In: Signale aus Köln: Beiträge zur Musik der Zeit, vol. 22. Der Apfel, Wien (2017)
17. Frisius, R.: Musique concrete*. http://www.frisius.de/rudolf/texte/tx355.htm*, retr. 21/06/14.
18. Eimert, H., Humpert, H.U.: Das Lexikon der elektronischen Musik. Bosse (1973).
19. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In: Proc. of the Int. Conf. on Learning Representations, 2015, pp. 1–14 (2015).
20. Hershey, S., Chaudhuri, S., Ellis, D., Gemmeke, J., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R., Seybold, B., Slaney, M., Weiss, R., and Kevin Wilson: CNN Architectures for Large-Scale Audio Classification. In: Proceedings of ICASSP 2017, pp. 131-135 (2017).

Knowledge Transfer from Neural Networks for Speech Music Classification

Christian Kehling[1,2] and Estefanía Cano[3] ⋆

[1] Institute for Digital Media Technology
Technical Univeristy of Ilmenau
[2] Fraunhofer Institute for Digital Media Technology
[3] Songquito UG
christian.kehling@tu-ilmenau.de

Abstract. A frequent problem when dealing with audio classification tasks is the scarcity of suitable training data. This work investigates ways of mitigating this problem by applying transfer learning techniques to neural network architectures for several classification tasks from the field of Music Information Retrieval (MIR). First, three state-of-the-art architectures are trained and evaluated with several datasets for the task of speech/music classification. Second, feature representations or embeddings are extracted from the trained networks to classify new tasks with unseen data. The effect of pre-training with respect to the similarity of the source and target tasks are investigated in the context of transfer learning, as well as different fine-tuning strategies.

Keywords: Deep Learning, Neural Networks, Audio Classification, Speech Music Classification, Transfer Learning, Embeddings, Music Information Retrieval

1 Introduction

Detection of speech and music in audio signals has been investigated in the field of Music Information Retrieval (MIR) to automatically enrich audio archives with metadata. In addition to binary classification where only one of the classes is assumed to be present at time more complex tasks like segmentation of speech or music as well as multi-label classification where multiple classes can be present at time gained popularity. Despite the vast amount of research in this field [23, 12, 14, 24, 13, 5, 20, 4, 8], speech/music classification (SMC) remains challenging in the presence of noise, the involvement of chanting, or under low-quality recording conditions [15]. SMC was first addressed with algorithms based on audio features (e.g., pitch, zero crossing rate) [23, 14, 12]. Recent approaches almost entirely focus on deep neural networks (DNN) that directly learn to detect desired audio properties from input signals and its corresponding annotations [13, 2, 5, 20]. In an attempt to make audio classifiers more robust to varying signal conditions and data scarcity, pre-trained feature representations (embeddings) from related tasks are tranferred to new tasks, so called Transfer Learning (TL), to avoid exhaustive training from scratch [3, 6, 8, 9, 2].

⋆ This work has been supported by the German Research Foundation (BR 1333/20-1, CA 2096/1-1)

This work is divided in two stages. First, we analyze three state-of-the-art neural network architectures for SMC and evaluate their robustness to varying signal conditions by using a diversity of datasets. Here we aim to understand whether any of the three architectures is more robust to varying signal characteristics when trained under comparable conditions. In the second stage of our work, audio embeddings are computed from the three pre-trained architectures. These embeddings are then transferred to different MIR tasks. In this stage, we aim to understand how pre-trained models compare to baseline networks trained from scratch, and whether a close relation of a downstream task and pre-training task exhibit higher learning effects than general audio embeddings like OpenL3 [3] that were not trained on a related MIR task at all.

2 Related Work

Current approaches for SMC mostly rely on deep neural networks (DNN) trained and optimized using raw audio data or its time-frequency transform. The most popular networks for this task are convolutional neural networks (CNN) [12, 13, 5, 20]. In 2015 Lidy et. al [13] used a CNN approach consisting of one convolutional layer followed by a fully connected layer achieving 99.7% accuracy on binary classification of speech and music at the MIREX competition [18]. The separate detection of both classes still achieved 88.5% accuracy. The model proposed by Marolt [15] obtained an accuracy of 98% for SMC, and 92% for a 4-class classification for speech, solo singing, choir, and instrumental music. The model uses a combination of convolutional layers followed by residual layers. Besides the `GTZAN` [25] and `MUSAN` [24] datasets, additional field recordings and traditional music from various libraries were included. In [4], different architectures including DNNs, CNNs and recurrent neural networks were evaluated for speech music detection. According to their findings, a model with six CNN layers performed best on AudioSet [21] with 86% accuracy for speech or music detection. SwishNet [8] uses a set of one-dimensional convolutions with multiple skip connections on Mel-Frequency Cepstral Coefficients (MFCCs). This model achieved 93% accuracy on a 3-class detection task with speech, music, and noise and 99% accuracy for speech detection using the `MUSAN` [24] dataset for training and `GTZAN` [25] for verification. For performance comparison Hussain et al. used a Gaussian Mixture Model, a fully connected neural network (FCN), and a transfer learning approach of the MobileNet architecture [7] was used. The MobileNet embeddings worked best throughout the paper followed by the proposed SwishNet architecture.

Choi et al. [2] showed that transfer learning can outperform traditional feature based methods in many different MIR tasks as well as audio event detection (AED). In [3] OpenL3 embeddings were trained on the task of audio-video correspondence in a self-supervised manner inspired by [1] and subsequently transferred to the task of environmental sound classification. On several AED datasets this approach outperformed other TL embeddings based on VGG-like and SoundNet architectures. Grollmisch et al. [6] verified the potential of OpenL3 for different MIR and industrial sound analysis tasks. The embeddings consistently resulted in good classification performance while other embeddings highly varied depending on the task. Kong et al. [11] proposed pre-trained audio neural networks (PANN) for transfer learning. The authors introduce an input rep-

resentation called Wavegram, a neural network based time-frequency-transformation. A multi-layer CNN is connected to this input network and trained for audio tagging on the AudioSet [21]. Subsequently, these embeddings were augmented by trainable classifiers and applied to six different classification tasks including genre and acoustic scenes classification, among others. In most of these tasks, the embeddings performed better or similar to state-of-the-art approaches. The authors compared multiple networks and depths as well as different positions for unfreezing of the pre-trained embeddings concluding that a complete fine-tuning of all network parameters results in the highest accuracy. To overcome the overfitting to one particular task Kim et al. [9] proposed multi-task learning. During training, a CNN network structure is split at one stage in the model into multiple branches, one for each task. All branches consist of the same network architecture and where trained simultaneously. The last layers before the classifiers of each branch are concatenated and used as combined embeddings. Initially the system was trained on the Million Song Database [16] for tempo estimation and song similarity. The embeddings were evaluated on target tasks like genre classification or music recommendation. Different branch positions in the network were evaluated concluding that earlier branching results in better performance for the target tasks but also in bigger networks with more computational costs.

3 Datasets

To get a better understanding of the performance of the evaluated architectures, four datasets were used during training as depicted in table 1. The `MUSAN` dataset [24] and the `GTZAN` dataset [25] consist of clearly distinguishable broadcast material of western music and speech. In addition, two more challenging ethnomusicology datasets are included. The `Marolt19` dataset was first introduced in [15]. Apart from the speech class, choir, solo singing and instrumental music are combined into the 'music' class for training. `Marolt19` includes material from archives such as the British Library world & traditional music collection, the French Centre of Scientific Research (CNRS), or the Slovenian sound archive Ethnomuse. The `ACMus Youtube Dataset (ACMusYT)`[4] was collected as part of the ACMus research project. [5] It consists of audio excerpts of

[4] `https://zenodo.org/record/4870820`

[5] ACMus project page: `https://acmus-mir.github.io/`

Table 1. Characteristics of the datasets used for training on speech/music classification (source task) and for transfer learning tasks (target tasks).

Application	Dataset ID	Classes [Number of Files per class]	Sample Rate	Bit Depth	Duration [min]
Training	MUSAN	Music [660], Speech [426], Noise [764]	16 kHz	16	6483
	GTZAN	Music [64], Speech [64]	22 kHz	16	64
	Marolt19	Solo Singing [1512], Choir [1618], Instrumental [2960], Speech [1284]	44 kHz	16	577
	ACMusYT	Speech [40], Music [35], A Cappella [40]	48 kHz	16	88
Transfer	S&S	Music [101], Speech [80]	22 kHz	16	45
	ACMusVF	Male [46], Female [24]	96 kHz	24	26
	ACMusIF	1 [43], 2 [42], 3 [43], 4 [21], 5+ [36]	96 kHz	24	65

traditional Colombian music from the Andes region. The subset used in this work consists of two classes: speech and music with vocals. The 'vocal-only' class is not used in these experiments for better separation during training. For TL experiments, the pretrained networks are subsequently fine-tuned with separate datasets. An established set for speech music tasks is the Slaney & Scheirer dataset (S&S) [23] with content taken from broadcast material. All 64 files of noise and mixed (speech/music) content are excluded before the evaluation. From the `ACMus-MIR` dataset [17], the `Instrumental Format Set (ACMusIF)` was used. This set was created from traditional Andean music recordings for the purpose of ensemble size classification. The goal of this task is to classify music tracks as solo, duo, trio, quartet, and larger ensembles. Finally, the `ACMus Vocal Format Set (ACMusVF)` is included.[6] It comprises Andean vocal music (male and female singers) partly with accompaniment.

4 Methodology

4.1 Network Architectures

The INA (Institut National de l'Audiovisuel) approach [5] is a CNN-based network that uses 68 frames of 21 MFCCs with a maximum frequency of 4 kHz as input representation to four 2D-convolutional layers followed by four dense layers with dropout. Each of these layers are followed by batch normalization and a $ReLU$ activation. The output layer uses $Softmax$ activation (see Figure 1 for details). INA achieved an average accuracy of 92.6% at the 2018 MIREX [19] competition on music detection and 96.2% on speech detection.

SwishNet is an architecture based on one-dimensional convolutional layers in combination with residual and skip connections [8] (see Figure 2). As input, 16 frames of 22 MFCCs are extracted from one second audio snippets and used as 2D feature representation. Classification results range from 93% frame-wise accuracy for 3 classes (speech, music, noise) to 99% segment-wise accuracy for speech detection.

VGG-like architectures are commonly used networks in many fields of deep learning [15, 3, 2]. The network illustrated in Figure 3 is inspired by [22]. Logarithmic MelSpectrogram (MelSpec) is used as input from audio sampled at 22050 Hz. Frames of 2048 samples with 512 samples hop size are transformed to 128 mel band representation. A patch of 10 frames is fed to four convolutional layers with 32 kernels of size 3x3.

[6] `https://zenodo.org/record/4791394`

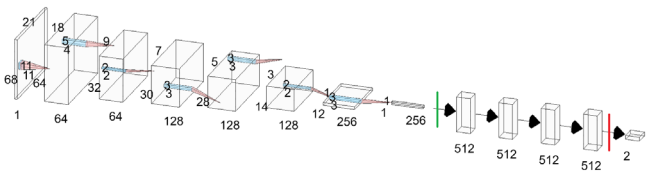


Fig. 1. INA network architecture [5]. The green line indicates the freezing point of the intermediate fine-tuning strategy. The red line indicates the output point of the embedding vector.

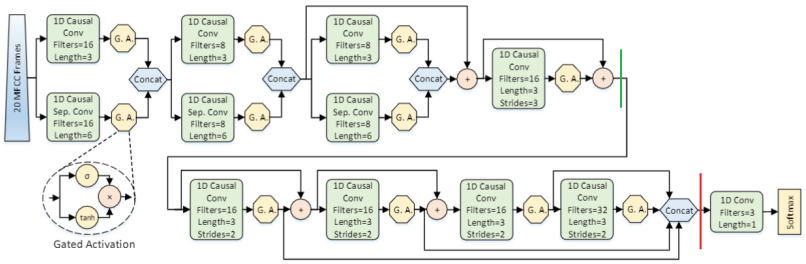


Fig. 2. SwishNet network architecture. The green line indicates the freezing point of the intermediate fine-tuning strategy. The red line indicates the output point of the embedding vector. Refer to [8] for more details on the architecture.

Each layer is followed by batch normalization and ReLU activation. After every second convolutional layer, MaxPooling is applied with a 3x3 window. Two fully connected layers are added after flattening followed by the classifier with a $Softmax$ activation.

OpenL3 embeddings are included as a state-of-the-art baseline. The 512 unit feature vectors are extracted from the audio data with default parameters from [3]. These vectors are normalized between 0 and 1 and used as input for a trainable neural classifier consisting of a 128 unit dense layer followed by the final classifier with $Sigmoid$ activation. As a second baseline, a simple DNN architecture is used. MelSpecs with equal measures as for SwishNet and VGG-like models are input and passed through one dense layer with 128 units and the output layer. The same structure is used for the appended classifiers of the computed embeddings in Section 4.4 and hence gives an insight into the learning effects of the preceded architectures. $Adam$ is used as optimization and $Softmax$ as activation function.

4.2 Input Representation

All datasets were normalized in a range of [-1, 1] in time domain and unified to a sampling rate of 22050 Hz and 16 bits. The MelSpec representation with 128 bands and 512 hop size is evaluated as input representation for all networks. Additionally the original MFCC input representations of the SwishNet and INA approach are included to check for side effects of the input adaption. The original VGG-like approach already

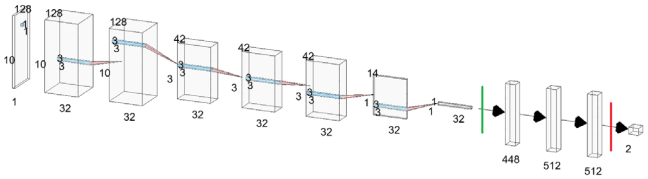


Fig. 3. VGG-like network architecture. The green line indicates the freezing point of the intermediate fine-tuning strategy. The red line indicates the output point of the embedding vector.

used MelSpecs. The OpenL3 embeddings create batches of features with a feature size of 512 samples (see Section 4.1) from 100 ms audio frames.

4.3 Implementation Details and Metrics

In all experiments, 10% of the data is used for testing, and 10% for validation. All experiments are repeated using five-fold cross-validation. All data is balanced by random down-sampling. After transforming the input to MelSpec, it is normalized feature-wise to zero mean in the range from -1 to 1 and concatenated to batches of 64 frames. Each network is trained for 200 epochs with the option for early stopping if the validation accuracy does not increase for 50 epochs. The $Adam$ optimizer [10] with a learning rate of 10^{-3} is used for all architectures for best comparability to the original implementations. Results are presented as the mean accuracy over 5 cross-validation folds with its standard deviation.

4.4 Transfer Learning Networks and Tasks

For transfer learning, the models are trained with a balanced combination of all four training sets. Afterwards the output layers are removed from the trained networks (see Section 4.1) and the remaining layers are fixed and used for embedding calculation. A trainable classifier is appended consisting of a 128 unit dense layer and a dense output layer matching the number of the target task classes. Three different freezing positions for the trained models are evaluated. In the first strategy, only the classifier is trained while the network weights remain fixed. The second strategy unfreezes the networks in an intermediate position so the classifier and parts of the networks are fine-tuned. These positions are illustrated green in Figures 1, 2, and 3, respectively. In a third strategy, all network weights are unfrozen and fine-tuned along with the classifier. These strategies do not apply for OpenL3 because of its baseline function. As transfer learning tasks, we evaluate the following target tasks: (a) SMC with `S&S` dataset, (b) accompaniment detection with `ACMusVF` dataset. The goal of this task is to distinguish music pieces with instrumental accompaniment from vocal-only performances, (c) female vs male singer classification on the `ACMusVF` dataset. We refer to this task as gender classification in singing, (d) ensemble size classification on the `ACMusIF` set.

5 Results

5.1 Network Architectures Comparison

Figure 4 shows the mean file-wise and frame-wise results over all training sets for each architecture. Results show that OpenL3 embeddings work well on all datasets for SMC. Looking at the frame-wise accuracy, SwishNet is slightly below the remaining two CNN-based architectures by around 3%. Figure 5 presents results for binary SMC and a three-class task which includes noise as the third class. This is performed for the `MUSAN` and `Marolt19` datasets where noise samples are included. `Marolt19` appears to be the most challenging set due to the fact that it does not only consist of

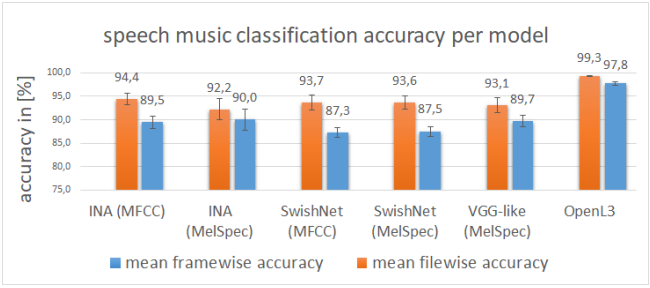


Fig. 4. Comparison of the mean frame-wise accuracy per architecture for speech/music classification averaged over all training sets (`MUSAN, GTZAN, Marlot19, ACMusYT`).

broadcast material unlike `MUSAN`. As expected, the accuracy drops for a more complex task of three classes. The highest drop of 24.3 % occurs for INA in connection with MelSpec input followed by the VGG-like model. For `MUSAN` the most significant drop can be observed for the INA model in connection with MFCC input. The varying results indicate that the INA architecture might not be well suited for alternative tasks in contrast to OpenL3 which shows best robustness. Regarding the input representation no significant performance differences can be observed in Figure 4. Only a slight improvement for MelSpecs is visible. Figure 5 confirms this trend as MelSpecs have a slightly better performance on average. In conclusion MFCCs can increase performance for specific tasks but MelSpecs have a more robust behavior in general hence MelSpec is used for further experiments.

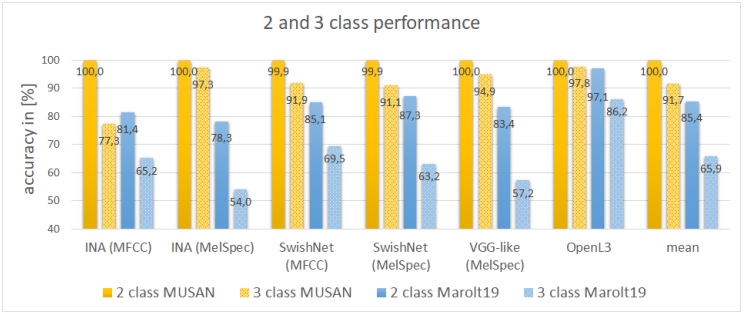


Fig. 5. Comparison of frame-based accuracy for binary classification versus 3-class classification. Results are shown for `MUSAN` (yellow) and `Marolt19` (blue) datasets.

5.2 Transfer Learning

Results for all transfer experiments are presented in Table 2. Besides the three network architectures (INA, SwishNet, and VGG-like), results for the OpenL3 embeddings and

the DNN baselines are shown. In general the resulting models tend to overfit during fine-tuning due to the small training data.

Speech Music Classification with S&S: In this experiment, the target task for TL was kept the same so models are only transferred to an unseen dataset. In Table 2 a learning effect from the pre-training can be observed for the Slaney & Scheirer dataset. In detail embeddings from INA and VGG-like models can make better use of pre-training and gain up to 3 % classification accuracy while the performance of SwishNet remains at almost the same level. OpenL3 embeddings outperform all other models for this dataset-task combination.

Table 2. Transfer learning results. Accuracy values are presented for fully frozen (Acc_{FZ}), partly trainable (intermediate) (Acc_{IN}), and the fully trainable embeddings (Acc_{FT}). Listed are the results for each architecure using their pre-trained embeddings (*Emb*) as well as their original network trained from scratch on the according task (*Orig*). In addition OpenL3 embeddings and the two-layer DNN (see 4.1) are listed as baseline.

Task-Set-Combination	Model	Acc_{FZ} [%]	Acc_{IN} [%]	Acc_{FT} [%]
Speech Music on S&S	INA_{Emb}	98,8 ± 1,4	97,6 ± 2,1	85,1 ± 4,8
	INA_{Orig}	-	-	93,8 ± 3,0
	$VGG-like_{Emb}$	97,4 ± 1,5	97,9 ± 1,9	88,9 ± 1,6
	$VGG-like_{Orig}$	-	-	95,1 ± 2,1
	$SwishNet_{Emb}$	92,3 ± 2,7	93,0 ± 2,4	95,0 ± 1,7
	$SwishNet_{Orig}$	-	-	92,9 ± 1,5
	$OpenL3_{Emb}$	**99,2** ± 0,4	-	-
	$DNN_{baseline}$	-	-	92,9 ± 1,9
Accompaniment on ACMus VF	INA_{Emb}	85,2 ± 5,6	82,5 ± 9,1	90,8 ± 5,2
	INA_{Orig}	-	-	80,2 ± 6,6
	$VGG-like_{Emb}$	88,5 ± 7,4	94,9 ± 3,2	92,7 ± 5,8
	$VGG-like_{Orig}$	-	-	92,7 ± 4,9
	$SwishNet_{Emb}$	81,5 ± 4,6	85,1 ± 4,6	93,6 ± 3,2
	$SwishNet_{Orig}$	-	-	94,0 ± 3,7
	$OpenL3_{Emb}$	**99,6** ± 0,5	-	-
	$DNN_{baseline}$	-	-	96,5 ± 1,7
Gender on ACMus VF	INA_{Emb}	70,0 ± 7,7	47,3 ± 7,9	59,3 ± 7,6
	INA_{Orig}	-	-	67,4 ± 7,0
	$VGG-like_{Emb}$	71,8 ± 5,2	75,8 ± 9,1	73,5 ± 6,2
	$VGG-like_{Orig}$	-	-	73,6 ± 8,1
	$SwishNet_{Emb}$	72,6 ± 5,0	73,1 ± 5,1	**78,3** ± 8,9
	$SwishNet_{Orig}$	-	-	74,9 ± 9,5
	$OpenL3_{Emb}$	72,3 ± 9,6	-	-
	$DNN_{baseline}$	-	-	72,6 ± 10,3
Ensemble Size on ACMus IF	INA_{Emb}	49,8 ± 5,6	52,1 ± 10,2	56,7 ± 4,5
	INA_{Orig}	-	-	48,8 ± 7,2
	$VGG-like_{Emb}$	49,7 ± 5,0	51,3 ± 6,8	47,1 ± 3,9
	$VGG-like_{Orig}$	-	-	57,9 ± 5,3
	$SwishNet_{Emb}$	46,7 ± 5,7	48,7 ± 6,3	54,3 ± 5,4
	$SwishNet_{Orig}$	-	-	56,3 ± 5,6
	$OpenL3_{Emb}$	**76,2** ± 4,4	-	-
	$DNN_{baseline}$	-	-	61,4 ± 5,3

Accompaniment detection on ACMusVF: For this task OpenL3 again shows best results and is followed by the VGG-like embeddings with a performance gap of around 11 %. Despite the close task relation to SMC no architecture overcomes the accuracy of the plain DNN and hence no learning effect from TL is achieved in connection with

this task. This is reinforced by the fact that for SwishNet and VGG-like architectures, the original models perform better than their embedding counterparts.

Female/Male singer classification on ACMusVF: For this task SwishNet embeddings show best results closely followed by OpenL3 embeddings. The original networks for each model show comparable or better performances compared to the fully frozen embeddings indicating that no learning effect of pre-training is visible. Again the DNN performs comparable to the best model refuting a benefit of the knowledge transfer.

Ensemble size classification on ACMus-MIR: All created embeddings perform similar with nearly 50 % accuracy. The baseline architectures of VGG-like and SwishNet show better results when trained from scratch excluding the idea of a possible learning effect. This is confirmed by the plain DNN baseline that outperformed the embeddings by around 12 %. The usage of embeddings results in a inverse effect for this task. Furthermore this experiment engages the most unrelated task relative to SMC in the set of transfer tasks. The best results are achieved using the unrelated OpenL3 embeddings with 76.2 %. A file-wise evaluation of OpenL3 results in 84 % accuracy which confirms the outcome from Grollmisch et al. [6].

Freezing strategies: Inspecting the last two rows of each embedding in table 2 gives insights to freezing strategies for the pre-trained networks. With more degree of freedom, meaning more trainable layers, the accuracy tend to increase in most cases. This trend is highly network-dependent and mainly applies to SwishNet models while INA tends to be more unstable showing a higher fluctuation. VGG-like models perform best in intermediate state.

6 Conclusions

This work examines the idea of transfer learning (TL) by creating new feature representations from one source task (pre-training), to use them as embeddings for several target MIR tasks. Three network architectures (INA, SwishNet, VGG-like) were initially trained for SMC, and subsequently applied to four new classification tasks. Our experiments show a slight dominance of the MelSpec as input representation over MFCCs during training. No significant performance difference between the three architectures is visible for the source task while OpenL3 embeddings consistently showed best SMC accuracy. In comparison to the networks trained from scratch, pre-training results in a slight improvement when used with an additional DNN classifier for the source task. In the TL experiments, the direct combination of MelSpec input and the DNN classifier surpasses the embedding performance in some cases. These results suggest that the learning effect of pre-training is not consistent over all experiments. Furthermore, creating embeddings with tasks closely related to the target tasks show no evident benefit compared to general audio embeddings such as OpenL3, which performed best in most of the cases. A possible cause can be the self-supervised creation of these embeddings which inhabits limitless availability of training data. However, the amount of training data used for pre-training the different embeddings is not considered in these experiments and is left for future work.

References

1. R. Arandjelovic and A. Zisserman. Look, listen and learn. *CoRR*, 2017.
2. K. Choi, G. Fazekas, M. Sandler, and K. Cho. Transfer learning for music classification and regression tasks. *CoRR*, 2017.
3. J. Cramer, H. Wu, J. Salamon, and J. Bello. Look, listen, and learn more: Design choices for deep audio embeddings. In *ICASSP*, 2019.
4. D. de Benito, A. Lozano-Diez, D. Toledano, and J. Gonzalez-Rodriguez. Exploring convolutional, recurrent, and hybrid deep neural networks for speech and music detection in a large audio dataset. *EURASIP*, 2019.
5. D. Doukhan, E. Lechapt, M. Evrard, and J. Carrive. Ina's mirex 2018 music and speech detection system. In *MIREX 2018*, 09 2018.
6. S. Grollmisch, E. Cano, C. Kehling, and M. Taenzer. Analyzing the potential of pre-trained embeddings for audio classification tasks. In *28th EUSIPCO*, 2021.
7. A. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and M. Andreetto. Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017.
8. S. Hussain and M. Ariful Haque. Swishnet: A fast convolutional neural network for speech, music and noise classification and segmentation, 2018.
9. J. Kim, J. Urbano, C. Liem, and A. Hanjalic. One deep music representation to rule them all? : A comparative analysis of different representation learning strategies. *CoRR*, 2018.
10. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.
11. Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. Plumbley. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *CoRR*, 2019.
12. A. Kruspe, D. Zapf, and H. Lukashevich. Automatic speech/music discrimination for broadcast signals. *LNI Proceedings*, 2017.
13. T. Lidy. Spectral convolutional neural network for music classification. *MIREX*, 2015.
14. M. Marolt. Probabilistic segmentation and labeling of ethnomusicological field recordings. *Proceedings of ISMIR*, 2009.
15. M. Marolt, C. Bohak, A. Kavcic, and M. Pesek. Automatic segmentation of ethnomusicological field recordings. *Applied Sciences*, 2019.
16. millionsongdataset.com. Welcome! — million song dataset.
17. F. Mora-Ángel, G. López Gil, E. Cano, and S. Grollmisch. ACMUS-MIR: An Annotated Dataset of Andean Colombian Music. In *7th DLFM Conference*, 2019.
18. music ir.org. 2015:music/speech classification and detection results - mirex wiki.
19. music ir.org. 2018:music and or speech detection results - mirex wiki.
20. M. Papakostas and T. Giannakopoulos. Speech-music discrimination using deep visual feature extractors. *Expert Systems with Applications*, 2018.
21. research.google.com. Audioset.
22. Y. Sakashita and M. Aono. Acoustic scene classification by ensemble of spectrograms based on adaptive temporal divisions. Technical report, DCASE2018 Challenge, 2018.
23. E. Scheirer and M. Slaney. Construction and evaluation of a robust multifeature speech/music discriminator. *ICASSP'97*, 1997.
24. D. Snyder, G. Chen, and D. Povey. Musan: A music, speech, and noise corpus, 2015.
25. G. Tzanetakis. marsyas.info gtzan speech music dataset download. `http://opihi.cs.uvic.ca/sound/music_speech.tar.gz`.

Three-Level Model for Fingering Decision of String Instruments

Gen Hori[1]

Asia University
hori@asia-u.ac.jp

Abstract. Fingering decisions of string instruments and other instruments differ in that the former involves string assignments as well as finger assignments while the latter is simply a matter of assigning fingers to notes. The present study introduces a three-level model for fingering decision of string instruments to describe the structure of the problem and present problem settings of fingering decision based on the model. Our proposed three-level model provides clear perspective for some problem settings of fingering decision. We perform a simulation to demonstrate the flexibility of the three-level model.

Keywords: fingering decision, string instruments, hidden Markov model(HMM)

1 Introduction

String instruments have overlaps in pitch ranges of their strings. As a consequence, they have more than one way to play even a single note and thus numerous ways to play a whole song. That is why the fingering decision for a given song is not always an easy task for string players and therefore automatic fingering decision has been attempted by many researchers. As for applications of HMM to fingering decision, Hori et al.[1] applied input-output HMM to guitar fingering decision and arrangement, Nagata et al.[2] applied HMM to violin fingering decision, and Nakamura et al.[3] applied merged-output HMM to piano fingering decision. Hori and Sagayama.[4] and Hori[5] proposed extensions of the Viterbi algorithm for fingering decision.

The purpose of the present study is to point out that fingering decisions of string instruments and other instruments differ in that the former involves string assignments as well as finger assignments while the latter is simply a matter of assigning fingers to notes. To describe the structure of fingering decision of string instruments, we introduce a three-level model for string instruments and provide a unified way of looking at variations of problem settings of fingering decision. Our proposed three-level model provides clear perspective for some problem settings of fingering decision. We perform a simulation to demonstrate the flexibility of our three-level model with fingering decision from score with finger numbers.

The rest of the paper is organized as follows. Section 2 reproduces the guitar fingering decision model based on HMM[1]. Section 3 points out the difference in fingering decision between string instruments and other instruments and introduces the three-level model. Section 4 presents problem settings of fingering decision based on the three-level model and Section 5 performs a simulation for one of the problem settings. Section 6 concludes the paper.

2 Fingering Decision Based on HMM

This section reproduces the guitar fingering decision model based on HMM[1] whose output symbols are musical notes and hidden states are left hand forms, which corresponds to the problem setting of Section 4.1 in this paper. Although we use the monophonic case as an example to simplify the explanation in the following sections, the results apply to the polyphonic case as well. See [1] for details of the polyphonic case.

2.1 HMM for fingering decision

To play a single note with a guitar, a guitarist depresses a string-fret pair p_i on fretboard,

$$p_i = (s_i, f_i),$$

with a finger h_i of the left hand and picks the string with the right hand. Therefore, a left hand form q_i for playing a single note can be expressed in a triplet q_i,

$$q_i = (s_i, f_i, h_i),$$

where $s_i = 1, \ldots, 6$ is a string number (from the highest to the lowest), $f_i = 0, 1, \ldots$ is a fret number, and $h_i = 1, 2, 3, 4$ is a finger number of the player's left hand (1,2,3 and 4 means the index, middle, ring and pinky fingers). The fret number $f_i = 0$ means an open string. The MIDI note number of the note played by the form q_i is calculated as follows where o_{s_i} denotes the MIDI note number of the open string s_i,

$$n(q_i) = o_{s_i} + f_i.$$

In this formulation, fingering decision is cast as a decoding problem of HMM where a fingering is obtained as a sequence of hidden states q_i given a score as a sequence of output symbols n_k.

2.2 Transition and output probabilities

The difficulty levels of the moves from forms to forms are implemented in the probabilities of the transitions from hidden states to hidden states; a small value of the transition probability means the corresponding move is difficult and a large value means easy. We assume that the four fingers of the left hand are always put on consecutive frets in this paper for simplicity. This lets us calculate the *index finger position* (the fret number the index finger is put on) of form q_i as $g(q_i) = f_i - h_i + 1$. Using the index finger position, we set the transition probability from hidden state q_i to hidden state q_j as

$$a_{ij}(d_t) \propto \frac{1}{2d_t} \exp\left(-\frac{|g(q_i) - g(q_j)|}{d_t}\right) \times P_H(h_j) \tag{1}$$

where $\propto$ means proportional and the left hand side is normalized so that the summation with respect to j equals 1 for all i. The first term of the right hand side is taken from the probability density function of the Laplace distribution that concentrates on the center

and its variance d_t is set to the time interval between the onsets of the $(t-1)$-th note and the t-th note. The second term $P_H(h_j)$ corresponds to the difficulty level of the destination form q_j defined by the finger number h_j.

As for the output probability, because all the hidden states have unique output symbols in our HMM for fingering decision, it is 1 if the given output symbol n_k is the one that the hidden state q_i outputs and 0 if the given output symbol is not,

$$b_{ik} = \begin{cases} 1 & (\text{if } n_k = n(q_i)) \\ 0 & (\text{if } n_k \neq n(q_i)) \end{cases} . \tag{2}$$

3 Three-Level Model for Fingering Decision of String Instruments

This section identifies the fundamental difference in fingering decision between string instruments and other instruments, and then introduces a three-level model for fingering decision of string instruments.

3.1 Note-tablature-form tree

For example, on the piano, there is only one key on the keyboard to press for each note, and therefore fingering decision for a given sequence of notes is a matter of deciding which finger to press on the key for each note (Fig.1, right). On the other hand, with the guitar, each note corresponds to several string-fret pairs that play it, and in addition, we have a matter of which finger to press for each string-fret pair (Fig.1, left). In other words, fingering decision for the piano is simply a matter of finger assignments, while fingering decision for the guitar consists of string assignments followed by finger assignments. This situation with the guitar is illustrated in a tree diagram (Fig.1, left) which we call "note-tablature-form tree." While the tree diagram in Fig.1 is for a monophonic note, we can draw the same diagrams for a polyphonic chord as well.

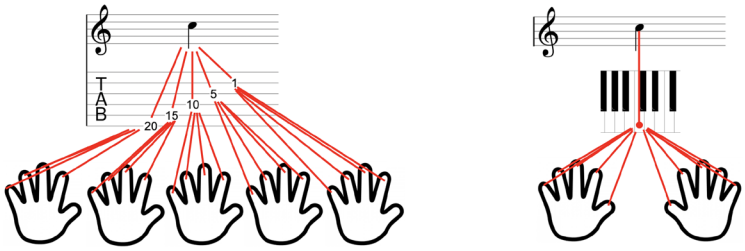


Fig. 1. Note-tablature-form tree for guitar (left) and corresponding diagram for piano (right) illustrating difference between string instruments and other instruments

3.2 Three-level model

To describe the above-explained situation with fingering decision of string instruments, we introduce a three-level model for string instruments that consists of (1) note level, (2) tablature level, and (3) form level (Fig.2). In relation to the notation introduced in Section 2.1, the note level contains the information of $n(q_i)$, the tablature level

$p_i = (s_i, f_i)$, and the form level $q_i = (s_i, f_i, h_i)$, respectively. In guitar scores, the score and the tablature contains the information of the note level and the tablature level, respectively. The finger numbers attached to the notes in the score, together with the tablature, make up the information of the form level (see Section 4.3). From the viewpoint of fingering decision based on HMM, the hidden states corresponds to the form level and the setting of observed symbols varies depending on the problem settings as we will see in the following sections.

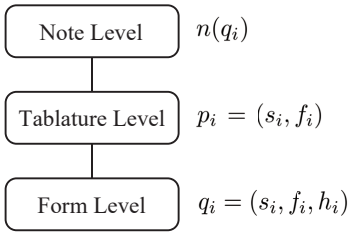


Fig. 2. Three-level model for fingering decision of string instruments

4 Problem Settings Based on Three-Level Model

This section provides a unified way of looking at variations in problem settings of fingering decision based on the three-level model for string instruments, taking the guitar as an example. Fingering decision is cast as a decoding problem of HMM where the setting of observed symbols varies depending on the problem settings. The first problem is a conventional one while the second and third ones obtain clear perspectives in light of our proposed three-level model.

4.1 Fingering decision from score

In this problem setting, we generate a sequence of forms from a score, taking the note level as the observed symbols and the form level as the hidden states (Fig.3, left). This is a conventional and common problem setting in guitar fingering decision and has been well studied including our previous study[1]. Here we note that the transition probability reflecting the difficulty of the form transition can be defined only in the form level and not in the tablature level, which we can see from the formula of transition probability (1). Even when we only need to generate tablature, we have to perform HMM decoding in the form level.

4.2 Fingering decision from tablature

In this problem setting, we generate a sequence of forms from a tablature, taking the tablature level as the observed symbols and the form level as the hidden states (Fig.3, right). Here we note that a tablature shows only string assignments for notes and does not contain information of finger assignments, although it is easy for skilled guitarists to find appropriate finger assignments and thus a fingering for a given tablature. An application example of this problem setting is difficulty assessment of a tablature where the difficulty is calculated as the reciprocal of the product of the transition probabilities along the generated sequence of forms.

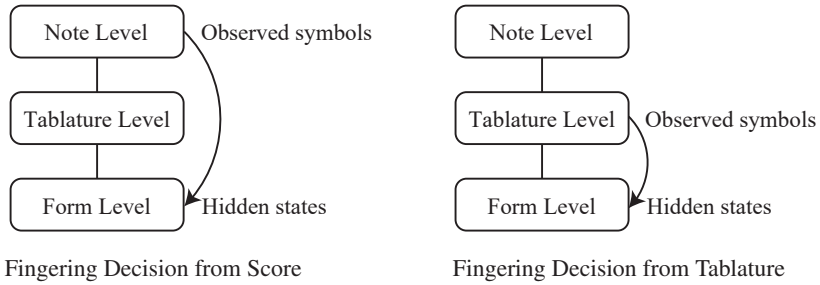


Fig. 3. Two problem settings based on three-level model

4.3 Fingering decision from score with finger numbers

There are guitar scores without tablatures with finger numbers attached to some key notes (Fig.4, left), which is enough for skilled guitarists to find a fingering for whole phrase. From the viewpoint of our proposed three-level model, this is a case where the whole information of the note level and the partial information of the form level are given to generate a sequence of forms. The fingering decision in this case is implemented as a decoding problem of HMM whose observed symbols are the notes and hidden states are forms limited to ones with indicated finger numbers. We will see some simulation results of this problem setting in the following section.

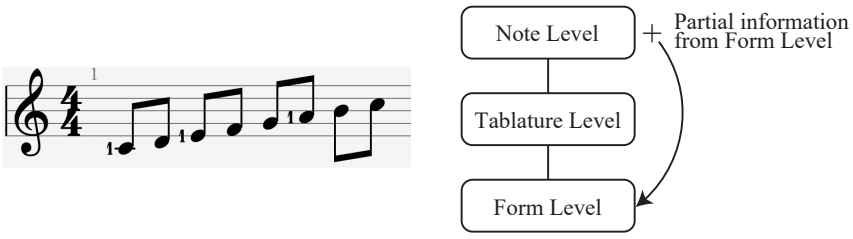


Fig. 4. Score with finger numbers (left) and corresponding problem setting (right)

5 Simulation

From the problem settings described in the previous section, we perform a simulation of one presented in Section 4.3 to demonstrate the flexibility of our proposed three-level model. The results for four scores are given in Fig.5 where the sequence of notes (C major scale) is common to all and the finger numbers with red circles are given while other finger numbers and the tablatures are generated by HMM. In the transition probability (1), we set $P_H(1) = 0.4$, $P_H(2) = 0.3$, $P_H(3) = 0.2$ and $P_H(4) = 0.1$ which means forms using the index finger are the easiest and the pinky finger the most difficult. From the results, we see that HMM generates appropriate fingerings for all the scores minimizing change in the index finger position and that specifying a finger number to one note can change fingerings for the rest seven notes.

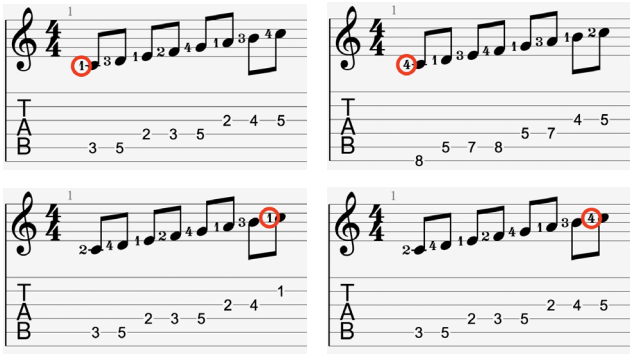


Fig. 5. Simulation results of fingering decision from score with finger numbers

6 Conclusion

We have pointed out the difference in fingering decision between string instruments and other instruments and introduced a three-level model for fingering decision of string instruments. Based on the model, we have provided a unified way of looking at three variations in problem settings of fingering decision and demonstrated the flexibility of our proposed three-level model using a simulation for fingering decision from score with finger numbers. There are other instruments than string instruments for which we have more than one way to play a single note. For such instruments, we can consider a fingering model with a middle level corresponding to the tablature level of our model for string instruments. We leave the extension of our three-level model to such instruments to our future study.

Acknowledgments. This work was supported by JSPS KAKENHI Grant Number 21H03462.

References

1. Hori, G., Kameoka, H., Sagayama, S.: Input-Output HMM Applied to Automatic Arrangement for Guitars. Journal of Information Processing, 21, 2, pp. 264–271. (2013)
2. Nagata, W., Sako, S., Kitamura, T.: Violin Fingering Estimation According to Skill Level Based on Hidden Markov Model. In: Proceedings of International Computer Music Conference and Sound and Music Computing Conference (ICMC/SMC2014), pp.1233–1238, Athens, Greece. (2014)
3. Nakamura, E., Ono, N., Sagayama, S.: Merged-Output HMM for Piano Fingering of Both Hands. In: Proceedings of International Society for Music Information Retrieval Conference (ISMIR2014), pp.531–536, Taipei, Taiwan. (2014)
4. Hori, G., Sagayama, S.: Minimax Viterbi Algorithm for HMM-Based Guitar Fingering Decision. In: Proceedings of International Society for Music Information Retrieval (ISMIR2016), pp. 448–453, New York City, U.S.A. (2016)
5. Hori, G.: Extension of Decoding Problem of HMM Based on L^p-Norm. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 3989–3993, IEEE. (2018)

Analysis of Musical Dynamics in Vocal Performances

Jyoti Narang, Marius Miron, Xavier Lizarraga, and Xavier Serra

Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
{jyoti.narang, marius.miron, xavier.lizarraga,
xavier.serra}@upf.edu

Abstract. Dynamics are one of the fundamental tools of expressivity in a performance. While the usage of this tool is highly subjective, a systematic methodology to derive loudness markings based on a performance can be highly beneficial. With this goal in mind, this paper is a first step towards developing a methodology to automatically transcribe dynamic markings from vocal rock and pop performances. To this end, we make use of commercial recordings of some popular songs followed by source separation and compare them to the karaoke versions of the same songs. The dynamic variations in the original commercial recordings are found to be structurally very similar to the aligned karaoke/multi-track versions of the same tracks. We compare and show the differences between tracks using statistical analysis, with an eventual goal to use the transcribed markings as guiding tools, to help students adapt with a specific interpretation of a given piece of music. We perform a qualitative analysis of the proposed methodology with the teachers in terms of informativeness and accuracy.

Keywords: Vocal Performance Assessment, Music Education, Loudness Measurement, Dynamics Transcription

1 Introduction

Musical expression is an integral part of any performance. The subjective nature of this term makes it difficult to identify "whether the expressive deviations measured are due to deliberate expressive strategies, musical structure, motor noise, imprecision of the performer, or even measurement errors" [1]. While the choice of expressions used may vary from performer to performer and also from performance to performance, deriving the expressions used in a specific interpretation of a performance can offer significant advances in the realm of music education. Not only can it help students learn from a specific musical piece, insights about the variations in expressions can add to possible set of choices that one can employ during a performance.

With the advent of online practice tools like music minus one, audio accompaniments, users have a wide variety of mediums to chose to practice with [2]. However, most of these tools are limited to pitch and rhythm correctness, offering little or no insight about the expressive variations of the performance. In this work, we focus on deriving the dynamic variations of vocal rock and pop performances via loudness feature extracted from the audio recordings. The goal of this paper is to develop a methodology to extract and compare the dynamic variations of similar pieces of vocal performances that can lay the foundation of transcribing dynamic markings of vocal performances.

This overall idea can be broken down into a set of 2 questions that we intend to address through our work.

(i) Given a mix, is it possible to transcribe dynamics using the source separated voice signal with the same accuracy as would be achieved when the vocal stem of the mix is available?

(ii) Can we analyze the similarities and differences between two loudness curves in order to provide feedback on dynamics?

In order to address the first question, we use state of the art source separation algorithms to extract vocal tracks from mixes followed by loudness computation, and compare them to the loudness curves of the vocal stems available for the same mix. To address the second question, we have conducted a preliminary experiment comparing the loudness curves of the source separated commercial mixes with multi-track karaoke versions with vocal stems. Overall the structure of the paper is as follows. Section 2 presents some fundamental information about the kind of loudness scales and the study of dynamics in music information retrieval. In section 3, we describe a methodology of the proposed approach followed by preliminary investigation of the comparison of loudness curves in section 4. The influence of vocal source separation on loudness computation is also presented in section 4.

In section 5, we conduct a case study where the dynamic variations of the two versions (karaoke and commercial) have been analyzed by a teacher to give feedback followed by section 6 with conclusions and future work.

2 Background and Related Work

Significant work has been done to model performance dynamics by measuring the loudness variations [3] with a conclusion that the variations in dynamics are not linear. Several measurement techniques have been defined to measure the loudness of signals.

2.1 Loudness Measurement Scales

Of the scales available for loudness measurement, some are inspired by the subjective psychoacoustic phenomenon of human ear, while others are objective in terms of measurement. The most commonly used measurement is the dBFS scale, or loudness unit full scale. The more recently adopted industry standard is the EBUR scale [9]. For our analysis, we make use of the sone scale, which is based on psychoacoustic model, and compare our results to RMS values computed from the signals directly.

Sone Scale This scale is inspired by the psychoacoustic concept of equal loudness curves, with the measurement being linear i.e. doubling of the perceived loudness doubles the sone value [10]. While the phon scale is more closely associated with dB scale, a phon value of 40 translates to 1 sone. The relationship between phons and sons can be modelled using the equation:

$$S = \begin{cases} 2^{(L-40)/10}, & \text{if } P >= 40. \\ (L/40)^{2.642}, & P < 40. \end{cases} \quad (1)$$

RMS RMS or root mean square is the square root of the mean square of the amplitude of the signal.

$$RMS = sqrt((x_1^2 + x_2^2....x_n^2)/N) \quad (2)$$

2.2 Dynamics in Music Information Retrieval

Work on measurement of dynamics has been typically centered around Western Classical piano performances, incorporating dynamics as an expressive performance parameter that can vary across performers/performances [4]. Kosta et al. [5] used change-point detection algorithm to measure dynamic variations from audio performances and compared them to the markings in the score. Further, they applied machine learning approaches like decision trees, support vector machines (SVM), artificial neural networks [6] to predict loudness levels corresponding to the dynamic markings in the score. They found that the loudness values can be predicted relatively well when trained across recordings of similar pieces, while failing when trained across pianists' other performances.

Another approach to model dynamics is using linear basis functions to encode structural information from the score [8]. Each of the "basis function" stand for one score marking like *stacatto*, *crescendo*, the active state being a representation of the expressive marking present in the score and vice-versa. Chacón et al. [7] carry out a large scale evaluation of expressive dynamics on piano and orchestral music using linear and non-linear models.

3 Methodology

A diagram of the proposed methodology is presented in Figure 1. In case solely the mix is available, the input audio mix is passed to a source separation algorithm, U-Net [16] to get the separated vocal track. Thereafter, we extract the loudness from the separated vocal track or vocal stem using the sone scale and RMS as described earlier. The loudness extraction for the sone scale is carried out in the same way as proposed by Kosta et al [5] in their analysis. Each of the loudness curves are normalized by dividing with the max value for the rendition in order to carry out a fair relative comparison between different renditions. This step makes sure that only the relative values are compared and not the absolute ones. Finally, we apply peak picking operation to get a range of overall dynamics that can be further processed to map to specific dynamics based on musicological knowledge. It is to be noted that we limit the current set of experiments to comparison of loudness curves, leaving the actual mapping of loudness values to musically meaningful values as future work.

4 Experiments

4.1 Data Curation

We have primarily used three sources of data for our analysis:

(i) Commercial official recordings of rock and pop songs

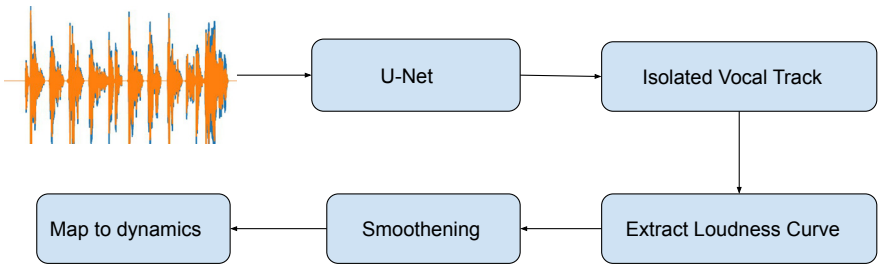


Fig. 1: Methodology for extracting loudness from a mix.

(ii) Custom karaoke tracks from the site[1] exactly replicating the official tracks
(iii) Musdb dataset to validate the efficacy of source separation algorithm

To evaluate the impact of singing voice source separation we use the musdb dataset containing 150 multi-track songs. For the commercial recordings, we conducted a preliminary investigation with 7 popular tracks shown in Table 1.

For the commercial popular recordings, only the mixes are available while for the karaoke versions, we have access to all the stems. This leads to 3 sources of data for the analysis of the same tracks - source separated vocals from the commercial mix (CSS), source separated vocals from the karaoke mix (KSS), vocal stems from the karaoke stems (KSV).

4.2 Experimental Setup

As mentioned above in the methodology, we first apply source separation using the spleeter implementation of UNet [13] to separate the mix into two stems - vocal track and the accompaniment. This step is skipped in case vocal stems are available for analysis. We use a block size of 512 samples or 11 ms with a hanning window, and a hop size of 256 samples or 5.5 ms. We follow the same block and hop size for the sone scale as well as RMS values. For loudness extraction using the sone scale, we use ma_sone function in Elias Pampalk's Music Analysis toolbox [11] in Matlab. The RMS values are extracted using the essentia library [15]. We further apply smoothening operation using two methods - "loess" with *smooth* function in matlab (based on locally weighted non-parametric regression fitting using a 2nd order polynomial) and exponential moving average [19][EMA]. Based on experimental testing, we use a span of 5% for the loess method. With the exponential moving average smoothening, we use an attack of 2 ms and release time of 20 ms. In the current set of experiments, the RMS smoothening is carried out using EMA methodology, and sone scale is smoothened using loess method. This operation was followed by peak picking operation to get a sense of overall dynamics followed. The peak picking parameters were experimentally set to a threshold of 0.1, and a peak distance of 1.2 seconds. We used the madmom library [14] for peak picking operation with RMS, and findPeaks function in maltab with sone scale loudness extraction. Figure 2 and Figure 3 show an example of computation of loudness

[1] https://www.karaoke-version.com/

value using Sone scale and RMS respectively, followed by smoothening operation and detected peaks for the song ‘Don’t know why’ by Norah Jones.

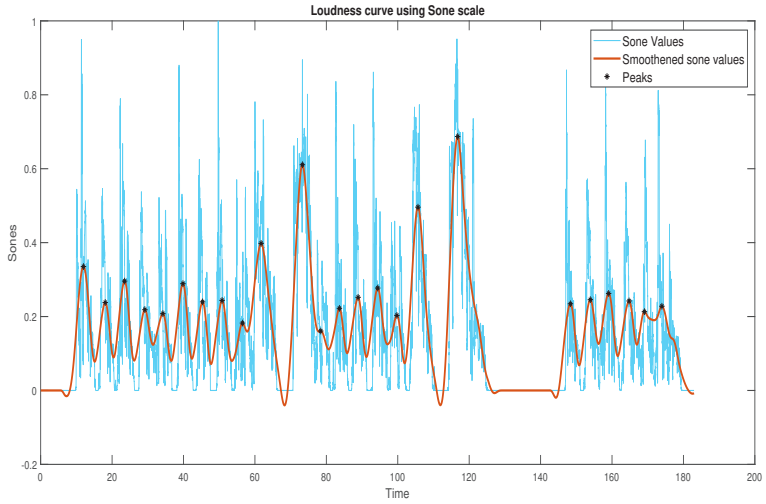


Fig. 2: Loudness using sone scale for Don’t Know Why by Norah Jones

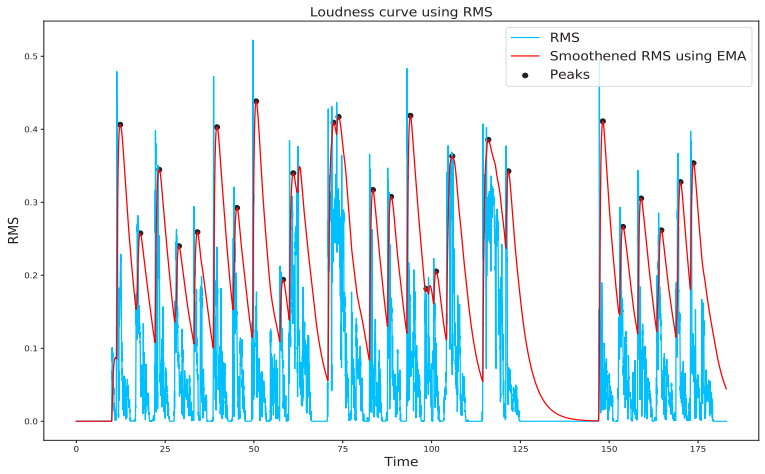


Fig. 3: Loudness using RMS values for Don’t Know Why by Norah Jones

4.3 Results

Overall Loudness Comparison between Renditions In order to compare the structure similarity of the loudness curves, we computed Pearson Correlation Coefficient of the smoothened curves extracted from the audio signals. Table 1 shows the values observed for each of the 7 songs. As evident from the table, most values are greater than 0.8, and in the case of comparing source separated version with the clean karaoke version, most values are greater than 0.9 indicating the robustness of the methodology with the pre-processing step of applying source separation.

Local dynamics To account for local dynamic changes, we compute the differences between consecutive peaks and derive a histogram from all the local differences. Further, the computed peak differences for each song are combined together for all songs from the same source i.e. commercial source separated, karaoke source separated and karaoke stem vocal. Thereafter, we use the non-parametric Kolmogorov-Smirnov 2 sample test which fits the properties of our data. This test is computed between each pair of the 3 histograms corresponding to the 3 sources. We find that for each of the comparisons, the p-value was 0.99 indicating no statistically significant differences between the histogram plots. These results are in line with our initial claim that the overall structure of the local dynamics changes as reflected in the loudness curves. These analysis results were the same for the histograms obtained using RMS values and sone values.

Table 1: Chosen songs and Pearson Correlation Coefficients for smoothened loudness sone curves

Song Name	**Artist**	**CSS, KSV**	**KSS, KSV**	**CSS, KSS**
Skyfall	Adele	0.867	0.994	0.931
Torn	Natalie Imbruglia	0.701	0.946	0.800
Fade into you	Mazzy Star	0.943	0.887	0.897
Imagine	John Lennon	0.889	0.981	0.440
Say you won't let go	James Arthur	0.955	0.835	0.800
Don't know why	Norah Jones	0.866	0.997	0.870
Son of a preacher man	Dusty Springfield	0.701	0.957	0.669

Global Dynamic Range The global dynamic range of each of the songs is computed using difference in max peak and min peak extracted from the smoothened loudness curve. As indicated in Table 2, the observed global dynamic range based on peak values are mostly similar in the case of karaoke source separated version and the karaoke vocal stem version with the exception of the song 'Son of a preacher man' with RMS values, and 'Fade into you' with sone values.

Outlier Analysis With a deeper analysis of the song 'fade into you', we find that there is a guitar section in the original song that becomes an artifact in the source separation output. This leads to a peak being wrongly detected increasing the overall dynamic

Table 2: Observed dynamic range with RMS and sone values

	RMS			Sone		
Song Name	**CSS**	**KSS**	**KSV**	**CSS**	**KSS**	**KSV**
Skyfall	0.460	0.156	0.176	0.503	0.477	0.489
Torn	0.092	0.138	0.206	0.355	0.199	0.213
Fade into you	0.144	0.195	0.167	0.306	0.354	0.182
Imagine	0.172	0.149	0.171	0.320	0.287	0.271
Say you won't let go	0.187	0.138	0.142	0.272	0.190	0.199
Don't know why	0.256	0.222	0.217	0.526	0.489	0.462
Son of a preacher man	0.150	0.227	0.371	0.275	0.339	0.295

range for both CSS and KSS resulting from peak detection. A high value of Pearson Correlation Coefficient between CSS and KSS as compared to KSS and KSV reflects from the fact that both of them have source separation as a pre-processing step, and both the versions contain similar artifacts.

4.4 Influence of voice source separation on loudness computation

In order to validate the efficacy of the source separation algorithm prior to using it for evaluating dynamics, we computed the Pearson Correlation of the smoothened loudness curves extracted from the mix with the smoothened loudness curves of the vocal stem tracks available with the musdb dataset [17].

As evident from the histogram in Figure 4, 138 values of the 149 songs evaluated are greater than 0.90. There are 6 songs with values between 0.80 and 0.90, and only 1 song with a value less than 0.50. The mean of the values is 0.960 and the standard deviation is 0.081. These results look promising to be able to use source separation as a prior step for dynamics analysis.

Outliers The song with the lowest value of correlation coefficient "PR-Happy Daze" contains a lot of instrumental music without much vocal component. Hence, the output of source separation algorithm is mostly artifacts. The song "Skelpolu - Resurrection" with a correlation coefficient of 0.58 has similar challenges.

5 Discussion

Work on transcription of dynamics is a challenging task for several reasons. One of the primary reasons being lack of sufficiently annotated data for singing voice to validate the efficacy of these algorithms.

Hence, in order to validate our approach, we conducted a case study with the song 'Don't know why by Norah Jones' where we asked a teacher with 6 years of Western singing teaching experience to compare the two tracks and provide feedback on the dynamic changes. Following is the feedback that we received from the teacher for some phrases of both tracks.

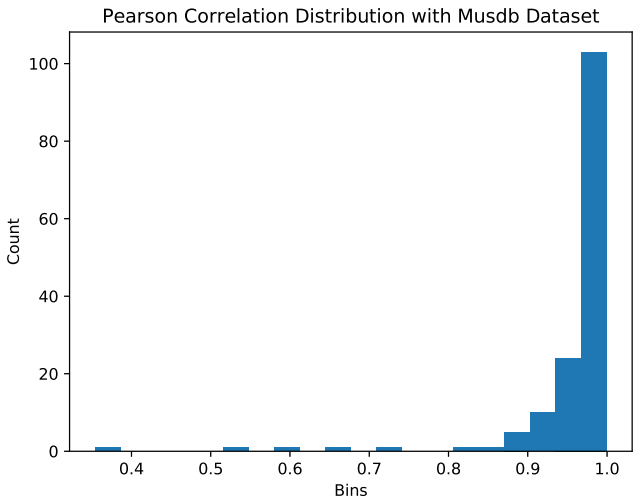


Fig. 4: Distribution of Pearson Correlation Coefficient applied to smoothened loudness curves of musdb dataset

I waited 'til I saw the sun

For Norah's Version:"Norah's dynamics change over the line. "I've" is 'mp'. "Waited till" starts as 'mf', which gradually drops down to 'mp' as she ends the line, can be seen as a diminuendo." For the Backing Track Version: "Dynamically, the singer is 'mf' throughout. This sounds like the kind of vocal take where the original vocals have been compressed one too many times."

I don't know why I didn't come

For Norah's Version: "Dynamically between an 'mp' and 'mf'". For the Backing Track Version: "Once again at an 'mf'. Vocals have definitely been compressed to sound at the same level consistently".

Case Study Results As evident from the first phrase, the teacher claimed that Norah Jones used a wider range of dynamics in her performance as compared to the cover version. Figure 5 shows the loudness curve of the cover version along with Norah Jones version using the sone scale. The classified dynamic markings for the two renditions are shown in the same plot. As compared to Norah's version of the same song, there is definitely a relatively very low difference between consecutive initial peaks in the cover version. The global dynamic range observed in the results section for this song is also in line with this observation. Similar results can be seen with RMS computation.

Challenges Despite having noisy artefacts and interferences from other instruments, state of the art source separation may be adequate for music analysis, when extracting dynamics. However, the peak detection method may not be robust enough to different

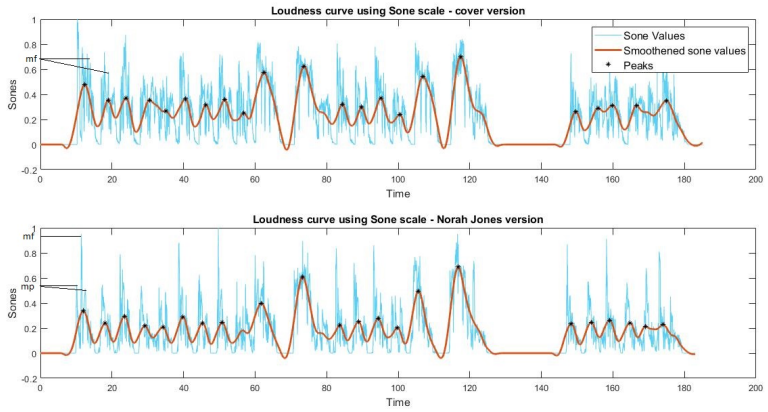


Fig. 5: Loudness using sone scale for Don't Know Why

performances and require calibration. Smoothing should be done w.r.t the tempo of the song.

While our initial case study showed some promising results, scaling such a system is still a very cumbersome task. Apart from the limitations with data and annotations, we are constrained by the knowledge that can help us realize the right granularity of transcription. For example, expressive markings like crescendo and diminuendo are associated with phrase boundaries [18], but the reverse might not be true. We would need collaborative efforts from multiple fronts in order to take advantage of the recent advances in the field of audio signal processing.

6 Conclusion and Future Work

We presented a methodology to extract dynamics from a performance using loudness as a feature. In the current investigation, we found that it is possible to use these loudness metrics to reach a level of relative changes that can in turn be mapped to dynamics. In future, we intend to discretise these relative values to map them to musically meaningful terms that can be used for providing the right feedback to students. Apart from that, in order to realize the overall goal of transcription, we intend to continue annotations of popular songs and further apply data driven approaches of machine learning to automatically derive the dynamic markings.

We also intend to apply the current methodology to student recordings to validate the efficacy of the system, and if the approach can be used to provide feedback on dynamics to students.

Acknowledgements We would like to thank Ajay Srinivasmurthy and Divakar Nambiath for their invaluable contributions to this work. Part of this research is funded by the projects Musical AI (PID2019-111403GB-I00/AEI/10.13039/501100011033 funded by the Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI)) and NextCore (RTC2019-007248-7 funded by the

Spanish Ministerio de Ciencia, Innovación y Universidades (MCIU) and the Agencia Estatal de Investigación (AEI)).

References

1. Langner, Jörg, and Werner Goebl. "Visualizing Expressive Performance in Tempo—Loudness Space." Computer Music Journal 27.4 (2003): 69-83.
2. Eremenko, Vsevolod, et al. "Performance assessment technologies for the support of musical instrument learning." (2020).
3. Berndt, Axel, and Tilo Hähnel. "Modelling musical dynamics." Proceedings of the 5th Audio Mostly Conference: A Conference on Interaction with Sound. 2010.
4. Widmer, Gerhard, and Werner Goebl. Computational models of expressive music performance: The state of the art." Journal of New Music Research 33.3 (2004): 203-216.
5. Kosta, Katerina, Oscar F. Bandtlow, and Elaine Chew. "Dynamics and relativity: Practical implications of dynamic markings in the score." Journal of New Music Research 47.5 (2018): 438-461.
6. Kosta, Katerina, et al. Mapping between dynamic markings and performed loudness: a machine learning approach." Journal of Mathematics and Music 10.2 (2016): 149-172.
7. Cancino-Chacón, Carlos Eduardo, et al. "An evaluation of linear and non-linear models of expressive dynamics in classical piano and symphonic music." Machine Learning 106.6 (2017): 887-909.
8. Grachten, Maarten, and Gerhard Widmer. "Linear basis models for prediction and analysis of musical expression." Journal of New Music Research 41.4 (2012): 311-322.
9. EBU–Recommendation, R. "Loudness normalisation and permitted maximum level of audio signals." (2011).
10. Beck, Jacob, and William A. Shaw. "Ratio-estimations of loudness-intervals." The American journal of psychology 80.1 (1967): 59-65.
11. Pampalk, Elias. "A Matlab Toolbox to Compute Music Similarity from Audio." ISMIR. 2004.
12. Kosta, Katerina. Computational Modelling and Quantitative Analysis of Dynamics in Performed Music. Diss. Queen Mary University of London, 2017.
13. Hennequin, Romain, et al. "Spleeter: a fast and efficient music source separation tool with pre-trained models." Journal of Open Source Software 5.50 (2020): 2154.
14. Böck, Sebastian, et al. "Madmom: A new python audio and music signal processing library." Proceedings of the 24th ACM international conference on Multimedia. 2016.
15. Bogdanov, Dmitry, et al. "Essentia: An audio analysis library for music information retrieval." Britto A, Gouyon F, Dixon S, editors. 14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8.. International Society for Music Information Retrieval (ISMIR), 2013.
16. Jansson, Andreas, et al. "Singing voice separation with deep u-net convolutional networks." (2017).
17. Rafii, Zafar, et al. "MUSDB18-a corpus for music separation." (2017).
18. Smith, Jeffrey C. Correlation analyses of encoded music performance. Stanford University, 2013.
19. Giannoulis, D., Massberg, M., & Reiss, J. D. (2012). Digital dynamic range compressor design—A tutorial and analysis. Journal of the Audio Engineering Society, 60(6), 399-408.

The Matrix Profile for Motif Discovery in Audio - An Example Application in Carnatic Music

Thomas Nuttall[1], Genís Plaja[1], Lara Pearson[2], and Xavier Serra[1]

[1] Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain
[2] Max Planck Institute for Empirical Aesthetics, Frankfurt am Main, Germany
thomas.nuttall@upf.edu, genis.plaja@upf.edu
lara.pearson@ae.mpg.de, xavier.serra@upf.edu

Abstract. We present here a pipeline for the automated discovery of repeated motifs in audio. Our approach relies on state-of-the-art source separation, predominant pitch extraction and time series motif detection via the matrix profile. Owing to the appropriateness of this approach for the task of motif recognition in the Carnatic musical style of South India, and with access to the recently released Saraga Dataset of Indian Art Music, we provide an example application on a recording of a performance in the Carnatic *rāga*, *Rītigauḷa*, finding 56 distinct patterns of varying lengths that occur at least 3 times in the recording. The authors include a discussion of the potential musicological significance of this motif finding approach in relation to the particular tradition and beyond.

Keywords: Musical Pattern Discovery, Motif Discovery, Matrix Profile, Predominant Pitch Extraction, Carnatic Music, Indian Art Music

1 Introduction and Related Work

Short, recurring melodic phrases, often referred to as "motifs", are important building blocks in the majority of musical styles across the globe. The automatic identification and annotation of such motifs is a prominent and rapidly developing topic in music information retrieval [1–4], playing a significant role in music analysis [5–7], segmentation [8–10] and development of musical theory [11–13]. No consensus exists on how this is best achieved, and indeed difficulty and differences in evaluation make it hard to contextualize the efficacy of a method outside of the task to which it is applied. A thorough review and comparison of approaches that handle symbolic music representations can be found in [1] and [4] however in this paper we focus on the much more common case of music without notation, extracting repeated motifs from audio.

Difficulty in working with raw audio for this task stems from the incredibly dense amount of information contained in audio signals, simultaneously clouding that which we might be interested in and providing a heavy workload for computational methods. A common method of reducing this complexity is to extract from the raw audio an object or feature set that captures the aspect of the music most relevant to the type of motif desired, and to subsequently compute some self-similarity metric between all subsequence pairs to group or connect similar sections [14, 15]. This could take the form of audio features such as Mel-frequency cepstral coefficients (MFCC) [16, 17]

or chroma [15, 18], rhythmic onsets [19, 20] or monophonic pitch [21, 22]. When performed successfully, it is the latter that provides an abstraction with the most information pertaining to the melody in audio. And with more recent advances in both predominant pitch extraction [23] and time series motif detection [24], we are afforded the opportunity to revisit the approach of predominant pitch extraction/self-similarity in computationally feasible time on relatively large time scales.

Certain musical styles are particularly suitable for this type of analysis: for example, those for which automated transcription is not yet possible, and where the symbolic to sonic gap is such that musically salient units may sometimes be better characterised by segments of continuous time series pitch data than by transcriptions. This is the case in Indian Art Music (IAM), including Hindustani and Carnatic styles. Automated motif detection in these traditions is a limited but active area of research. In the case of Carnatic music, *svaras* (notes) are coarticulated (merged) through *gamakas* (ornaments) [25]. This characteristic provides particular challenges for processes involving automated segmentation, and can even mean that different Carnatic musicians' annotations of the same phrase may vary subtly in places, with different degrees of symbolic detail being possible. This leaves motif detection through time series pitch data as one of the most viable and popular approaches to finding meaningful melodic units in the style [26–28].

In this paper we demonstrate an approach for the automated discovery of repeated motifs in audio: state-of-the-art source separation [31], predominant pitch extraction using the Melodia algorithm [23] and ultra-fast means of time series motif detection via the matrix profile [24]. Owing to the appropriateness of this approach for the task of motif recognition in Carnatic music, and with access to the recently released Saraga Dataset of IAM [32], we provide an example application, applying these existing methods in this tradition. All code is available on GitHub[3] with a Jupyter notebook walk through of both the generalized and IAM-specific code.

2 Dataset

We demonstrate our approach on an example recording from the Saraga dataset [32]. Developed within the framework of the CompMusic project[4] and openly available for research, Saraga comprises two IAM collections, representing the Hindustani and Carnatic traditions. Both collections comprise several hours of music with accompanying time-aligned expert annotations and relevant musical (e.g. *rāga*, *tāla*, form) and editorial (e.g. artist, work, concert) metadata. In this work we focus on a performance taken from the Carnatic collection, 168 of which contain separate microphone recordings of: lead vocal, background vocal (if present), violin, mridangam and ghatam (if present). However, since these tracks are recorded from live performance, the multi-track audios in the dataset contain considerable background leakage, i.e., are not completely isolated from the other instruments.

We access and interact with the Saraga dataset through the mirdata library [33]. This tool provides easy and secure access to the canonical version of the dataset, while load-

[3] `https://github.com/thomasgnuttall/carnatic-motifs-cmmr-2021/`
[4] `https://compmusic.upf.edu/`

ing and managing the dataset contents (audio, annotations and metadata) to optimize our research pipeline.

3 Methodology

The process consists of two stages (1) the extraction from audio of a vocal pitch track, which consists of a one-dimensional time series representing the main melodic line of the performance and (2) the use of self-similarity euclidean distance to identify likely candidates for repeated motifs in the main melodic line.

3.1 Predominant Pitch Extraction

The quality and consistency of the predominant pitch extraction is paramount. Given the shortage of training data and algorithms to extract the vocal pitch from Carnatic music signals, our raw audio recording is subject to three processing steps to arrive at a one dimensional time series of pitch values representing the main melodic line.

Isolating the Vocal Source Where possible we use the vocal track recording for analysis (still containing leakage from other instruments). If this is not available, the mix is used. For the isolation of voice from the background instruments (both in mixed and vocal tracks), we use Spleeter, which is a deep learning based source separation library which achieves state-of-the-art results on automatically separating vocals from accompaniment [31].

Extracting the Predominant Pitch Curve We use one of the most popular signal processing based algorithms for predominant pitch estimation from polyphonic music signals, the Melodia algorithm [23], applying an equal-loudness filter to the signal beforehand to encourage a perceptually relevant extraction. In the majority of studies attempting this task in IAM, Melodia has achieved consistent and viable results [26,28–30,34]. We use a time-step of 2.9ms for the extraction.

Post-Processing Two post-processing steps are applied to the pitch track. (1) Gap interpolation, linearly interpolating gaps of 250ms or less [36], typically caused by glottal sounds and sudden decrease of pitch salience in *gamakas* and (2) Gaussian smoothing with a sigma of 7, softening the curve and providing a more natural, less noisy shape.

The final extracted pitch track is a time-series of n pitch values, $P = p_1, p_2, ..., p_n$.

3.2 Repeated Motif Discovery

To search P for regions of similar structure we look for groups of subsequences that have a low euclidean distance between them. The subsequence length to search for, m is a user-defined parameter of the process.

Matrix Profile An efficient method of inspecting the euclidean distances between pairwise combinations of subsequences in a time series is the matrix profile [24]. Given a time series, T, and a subsequence length, m, the matrix profile returns for each subsequence in T, the distance to its most similar subsequence in T. The STAMP algorithm computes the matrix profile in impressive time by exploiting the overlap between subsequences using the fast Fourier transform, requiring only one parameter, subsequence length, m [24]. We use the *non-z-normalized* distance, since we are interested in matching subsequences identical in shape *and* y-location (i.e. pitch).

The matrix profile is therefore defined as $MP = ed_1, ed_2, ..., ed_{n-m}$ where ed_i is the regular euclidean distance between the subsequence of length m beginning at element i and its nearest neighbour in P.

Exclusion Mask To ensure that only subsequences of interest are considered, a mask of subsequences in P to exclude is computed by applying a series of *exclusion functions* to each subsequence. These exclusion functions are informed by expert understanding of what constitutes a relevant motif in the tradition. Explicitly, the exclusion mask, $EM = em_1, em_0, ..., em_n$ where em_i is either 1 or 0, yes or no, does the subsequence satisfy any of the following:

- *Too silent* - more than 5% percent of the subsequence is 0 (i.e. silence)
- *Minimum gap* - subsequence contains a silence gap of 250ms or more
- *Too stable* - in more than 63% of cases for a rolling window of 100, the average deviation of pitch from the average is more than 5 Hz. This step is designed to exclude subsequences with too many long held notes - although musically relevant, not interesting from a motific perspective. A similar approach is taken in [26]

Subsequences that correspond to a mask value of 1 are not considered valid and not returned.

Identifying Motif Groups The search for groups of repeated motifs begins by looking for a *parent* subsequence; those in P that have the lowest euclidean distance to another subsequence i.e. minimas in MP. The assumption being that if these subsequences have one very near neighbour, i.e. they are repeated once, then they are more likely to occur multiple times; a similar approach is used in [27].

For a candidate parent motif, we use the MASS similarity search algorithm [24] to calculate the non-normalised euclidean distance to every other subsequence in the pitch track, returning those that satisfy the requirements set by the parameters; $topN, maxOcc$, $minOcc$ and $thresh$. Algorithms 1 and 2 describe the process and parameters.

Output The returned motif groups are arrays of start indices in P. The number of groups and occurrences in each is influenced by the $topN$, $minOcc$ and $maxOcc$ parameters.

Algorithm 1 Identify groups of motifs with low inter-group euclidean distance

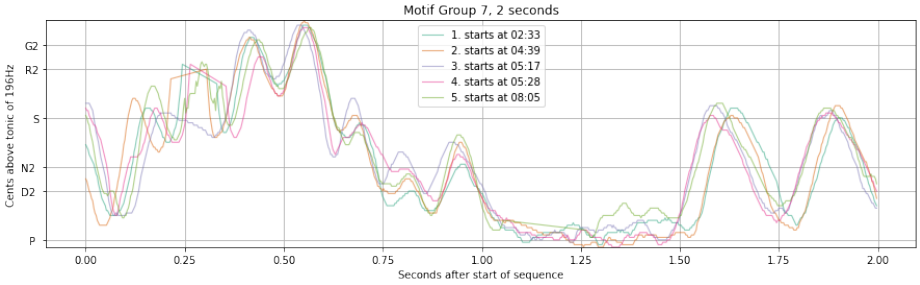
procedure GETMOTIFGROUPS
  MP ← matrix profile array from ***Matrix Profile***
  P ← pitch sequence array from ***Predominant Pitch Extraction***
  EM ← exclusion mask array from ***Exclusion Mask***
  m ← pattern length
  topN ← maximum number of groups to return
  maxOcc ← maximum number of occurrences per group
  minOcc ← minimum number of occurrences per group
  thresh ← maximum length-normalised distance of occurrence to parent

  MP[where(*EM* $== 1$)] $\leftarrow \infty$
  nGroups $\leftarrow 0$
  allMotifs $\leftarrow array()$
  while *nGroups* $<$ *topN*
    ix $\leftarrow argmin(MP)$ ▷ get parent index
    if *MP[ix]* $== \infty$ ▷ entire sequence searched
      break
    motifs ← GETOCCURRENCES(*ix*, *P*, *m*, *maxOcc*, *thresh*, *EM*)
    if Length(*motifs*) $<$ *minOcc* ▷ discard, not enough significant matches
      continue
    for *mtf* in *motifs* ▷ *motifs* is an array of indices
      MP[*mtf* - *m* : *mtf* + *m*] $\leftarrow \infty$ ▷ clear part of array to avoid future discovery
    nGroups ← *nGroups* $+ 1$
    allMotifs ← **append** *motifs*
  return *allMotifs* ▷ array of motif groups, each motif group an array of start indices
end procedure

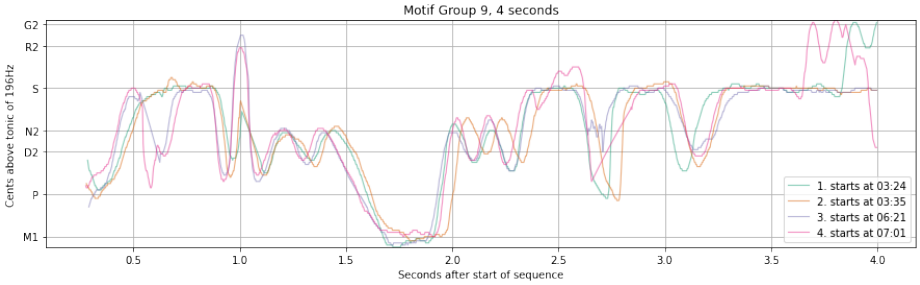
Algorithm 2 Identify other occurrences of parent motif in P using MASS

procedure GETOCCURRENCES
  ix ← index of parent sequence to query
  P ← pitch sequence array from **Predominant Pitch Extraction**
  m ← pattern length
  maxOcc ← maximum number of occurrences to return
  thresh ← maximum length-normalised distance of occurrence to parent
  EM ← exclusion mask array from **Exclusion Mask**

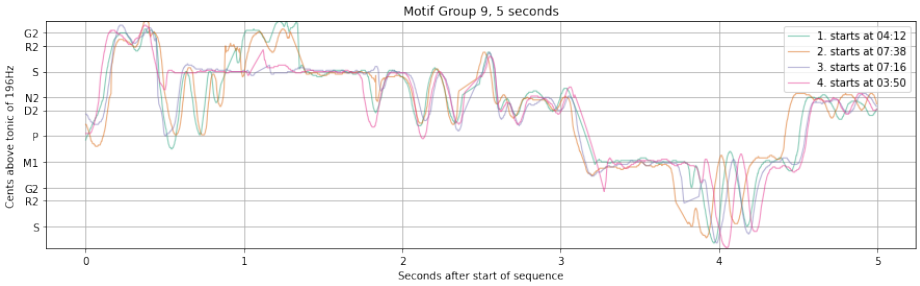
  parent ← *P*[*ix* : *ix* + *m*]
  stmass ← *MASS*(*parent*, *P*) ▷ array of distances between *parent* and all subsequences
  stmass[where(*EM* $== 1$)] $\leftarrow \infty$
  nOccs $\leftarrow 0$
  allOccs $\leftarrow array()$
  while *nOccs* $<$ *maxOcc*
    ix $\leftarrow argmin(stmass)$
    if *stmass*[*ix*]$/m >$ *thresh* ▷ length normalised distance
      break ▷ cease search, no significant patterns remain
    stmass[*ix* - *m* : *ix* + *m*] $\leftarrow \infty$
  allOccs ← **append** *ix*
  return *allOccs* ▷ array of occurrence start indices for this parent
end procedure



(a) Motif 7 - 2 seconds



(b) Motif 9 - 4 seconds



(c) Motif 9 - 5 seconds

Fig. 1: Overlaid pitch contour plots of three returned motif groups. The y-axis of each figure represents cents above the tonic (S) of 196Hz, divided into the discrete pitch positions defined in Carnatic music theory for this *rāga* - S, R2, G2, M1, P, D2, N2 [35]. R2 is two semitones (200 cents) above the tonic, S, and G2 is one semitone (100 cents) above R2, and so on. The oscillatory melodic movement that can be seen cutting across these theoretical pitch positions is typical of the style, illustrating the challenges of locating individual 'notes', either through expert annotations or automatically.

4 Results

We include the results of our process applied to a performance by the Akkarai Sisters of a composition titled Koti Janmani[5], by the composer Oottukkadu Venkata Kavi, which

[5] `https://musicbrainz.org/recording/5fa0bcfd-c71e-4d6f-940e-0cef6fbc2a32`

is set in the Carnatic *rāga*, *Rīt̄igauḷa*. The process is run for pattern lengths of 2,3,4, 5 and 6 seconds using parameters; $topN = 15$, $minOcc = 3$, $maxOcc = 20$. The parameter $thresh$ is selected by subjective evaluation of the patterns returned in one motif group, choosing a value beyond which consistency is lost.

The number of significant motif groups found for 2, 3, 4, 5 and 6 second runs is 15, 15, 11, 11 and 4 respectively. For the code and full results we refer the reader to the GitHub repository. Fig. 1a, 1b and 1c present the pitch plots associated with the top 5 occurrences of an example pattern in the 2, 4 and 5 seconds groups respectively.

5 Discussion

Due to the current lack of complete (i.e., saturated) ground truth annotations in the Saraga dataset, it is difficult to evaluate our application systematically. Creation of such annotations are ongoing as part of this project. In the meantime, however, the nature of the task and size of the results allow us to reflect on the coherency between patterns and their significance within the tradition.

The high degree of similarity between patterns returned within groups is obvious even to listeners who have no experience of the style, and can be appreciated from both the audio and pitch plots. This similarity is unsurprising, we choose a modest euclidean distance threshold and the process returns motifs that correspond to areas of pitch that are very similar by this measure. It is however a testament to the quality and consistency of the pitch extraction process and audio in the Saraga dataset [32], both resources not yet available in previous works. And more impressive still, also unseen in other works, is that these results can be achieved relatively quickly on a personal machine requiring little user input: pattern length, m and euclidean distance threshold, $thresh$ (easily tuned in negligible time). This is due to the efficiency of the STAMP and MASS algorithms in computing the all pairs self-similarity [24].

Of course, we are more interested in whether the consistent results identified by a process like ours have the potential to contribute to ongoing musicological endeavours of pattern recognition, documentation and music analysis in the Carnatic tradition. Initial evaluation by the third author, who has expertise in the tradition [25], suggests that that there is a high degree of musical similarity across the returned patterns in each group. At least the first few matches, and often all of the patterns, in each group would be considered by experts in the style to consist of the same motifs, or motif fragments. Some of the returned groups contain whole motifs that are particularly important for this *rāga*; *Rīt̄igauḷa* is one of the Carnatic *rāgas* that is expressed through a number of characteristic motifs, sometimes referred to as *pidi* (catch-phrases), *sañcāras* or *prayogas* [35].

Two examples of particularly musically significant motifs returned can be seen in Fig. 1a and Fig. 1b. Fig. 1a shows a frequently recurring phrase in this composition that includes the motif "npnn" (expressed here in *sargam* notation, which is used by practitioners to represent Carnatic *svaras*). The fact that 11 results are returned for this pattern (only five of these are illustrated for the sake of visual clarity) points to both the significance of the phrase in this composition, and also the importance of the motif "npnn" in the *rāga* [35]. Fig. 1b consists of another recurring characteristic phrase

"ssndmmnns", which is amongst the annotations of characteristic phrases identified by Carnatic musicians for the Saraga dataset [32].

The musicological applications of this process as it stands are limited to some extent by the fact that some of the matches returned are not full motifs, but rather are partial: for example, including part of one motif and then part of another (e.g., 5-second motif 0) or not returning the full motif (e.g., 5-second motif 1).[6] Segmentation at musically meaningful junctures such as silences or articulation of consonants should improve this. Another problem is that the process currently often returns multiples of the same motif, but with different top matches (e.g., 5-second motif groups 9 and 10). Lastly, it is clear that we need to evaluate the results against comprehensive annotations of all motifs in the performance,[7] to discover whether the process returns a good number of the total number of occurrences.

One interesting feature is that the process, in addition to returning precise matches of motifs, also identifies those that are similar but not identical. This could be particularly useful in a style such as Carnatic music which often employs a theme and variation structure, where phrases are repeated many times but with various elaborations. We can see an example of this returning of non-identical, but musically closely-related motifs in Fig. 1c where 4 motifs are returned, with two of them including a variation in the period between 0.5-1.5 seconds. Any process used to identify motifs in Carnatic music for musicological purposes would ideally show this degree of flexibility, in order to provide useful and meaningful results. Finally, considering the significance of recurring motifs in the vast majority of musical styles, it seems likely that this process would be musically relevant beyond the specific case of Carnatic music.

6 Further Work

Close scrutiny of the results offers potential lines of improvements; variable length motif detection could help capture full motifs rather than partial motifs, so too could more tradition-specific exclusion rules such as consonant onset detection, which should aid in further constraining the search to whole motifs due to the fact that the style is melismatic, with several *svaras* often sung to one syllable. An essential next step for the continuation of this work is the development of a more empirical evaluation framework of comprehensive ground truth motifs created in collaboration with expert performers of the tradition. We also recognize that to facilitate inter-recording discovery, a dynamic time warping distance measure or tempo normalisation might be necessary.

7 Conclusion

We hope to have demonstrated the effectiveness of predominant pitch extraction and matrix profile/self-similarity for the task of repeated motif identification and annotation in audio. We highlight its potential for these tasks in Carnatic music, a tradition where

[6] Please refer to the Github repository for results not plotted here.

[7] Although some motifs are annotated in the Saraga dataset, these annotations are not complete. Such annotating is extremely time consuming and must be done by practitioners of the style.

transcriptions into symbolic representation can show variance, and so where working directly with time series pitch data from audio is a more promising approach to motif identification. Alongside this document we provide the code and full results for the application to this tradition as well as to example audio from other musical styles.

Acknowledgments. This research was funded by the MUSICAL AI project (PID2019-111403GB-I00) granted by the Ministry of Science and Innovation of the Spanish Government. We also thank Rafael Caro Repetto for his continued guidance and input.

References

1. Ren, I. Y., Volk, A., Swierstra, W., Veltkamp, R. C.: A Computational Evaluation of Musical Pattern Discovery Algorithms. In: CoRR (2020)
2. 2017:Discovery of Repeated Themes & Sections - MIREX Wiki, `https://www.music-ir.org/mirex/wiki/2017:\%20Discovery_of_Repeated_Themes_\%26_Sections`
3. Ren, I. Y., Volk, A., Swierstra, W., Veltkamp, R. C.: In Search Of The Consensus Among Musical Pattern Discovery Algorithms. In: Proceedings of the 18th International Society for Music Information Retrieval ISMIR, pp. 671–680 (2017)
4. Janssen, B., Bas de Haas, W., Volk, A., van Kranenburg, P.: Finding repeated patterns in music: State of knowledge, challenges, perspectives. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8905, pp. 277–297 (2014)
5. Forth, J.: Cognitively-motivated geometric methods of pattern discovery and models of similarity in music. PhD thesis. Goldsmiths, University of London (2012)
6. Volk, A., van Kranenburg, P.: Melodic similarity among folk songs: An annotation study on similarity based categorization in music. In: Musicae Scientiae 16.3, pp. 317–339 (2012)
7. Ren, I. Y.: Closed Patterns in Folk Music and Other Genres. In: Proceedings of the 6th International Workshop on Folk Music Analysis, FMA, pp. 56–58 (2016)
8. Cambouropoulos, E.: Musical parallelism and melodic segmentation:: A computational approach. In: Music Perception 23.3, pp. 249–268 (2006)
9. Conklin, D., Anagnostopoulou, C.: Segmental pattern discovery in music. In: INFORMS Journal on computing 18.3, pp. 285–293 (2006)
10. Boot, P., Volk, A., Bas de Haas, W.: Evaluating the Role of Repeated Patterns in Folk Song Classification and Compression. In: Journal of New Music Research 45.3, pp. 223–238 (2016)
11. Nuttall, T., Casado, M., C., Ferraro, A., Conklin, D., Caro Repetto, R.: A computational exploration of melodic patterns in Arab-Andalusian music. In: Journal of Mathematics and Music, pp. 1-13 (2021)
12. Gjerdingen, R.: Music in the galant style. OUP USA (2007)
13. Rao, P., Ross, J. C., Ganguli, K. K.: Distinguishing raga-specific intonation of phrases with audio analysis. Ninaad, pp. 26–27(1), pp. 59–68 (2013)
14. Klapuri, A.: Pattern induction and matching in music signals. In: Exploring Music Contents. 7th International Symposium, CMMR, Málaga, Spain. pp. 188–204 (2010)
15. Dannenberg, R., B.: Pattern Discovery Techniques for Music Audio. In: Journal of New Music Research, 32 (2003)
16. Thomas, M., Murthy, Y., V., S., Koolagudi, S., G.: Detection of largest possible repeated patterns in Indian audio songs using spectral features, 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pp. 1-5 (2016)

17. Lie, L., Wang, M., Zhang, H.: Repeating pattern discovery and structure analysis from acoustic music data. In: Proceedings of the 6th ACM SIGMM International Workshop on Multimedia Information Retrieval, MIR, pp 275-282 (2004)
18. Wang, C., Hsu, J., Dubnov, S.: Music Pattern Discovery with Variable Markov Oracle: A Unified Approach to Symbolic and Audio Representations. In: Proceedings of the 16th International Society for Music Information Retrieval Conference, pp 176-182 (2015)
19. Krebs, F., Böck, S. Widmer, G.: Rhythmic Pattern Modeling for Beat and Downbeat Tracking in Musical Audio. In: Proceedings of the 14th International Society for Music Information Retrieval Conferences (2013)
20. Foote, J., Cooper, M., Nam, U.: Audio Retrieval by Rhythmic Similarity. In: Proceedings of the 3rd International Society for Music Information Retrieval Conference (2002)
21. Dannenberg, R., Ning, H.: Discovering Musical Structure in Audio Recordings. In: Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science) (2003)
22. Gulati, S.: Computational Approaches for Melodic Description in Indian Art Music Corpora. PhD Thesis, Universitat Pompeu Fabra, Barcelona (2016)
23. Salamon, J., & Gomez, E.: Melody extraction from polyphonic music signals using pitch contour characteristics. In: IEEE Transactions on Audio, Speech and Language Processing, pp. 1759–1770 (2012)
24. Yeh, C., M. et al.: Matrix Profile I: All Pairs Similarity Joins for Time Series: A Unifying View That Includes Motifs, Discords and Shapelets. In: IEEE 16th International Conference on Data Mining (ICDM), pp. 1317-1322 (2016)
25. Pearson, L.: Coarticulation and gesture: an analysis of melodic movement in South Indian raga performance. In: Music Analysis, 35(3), pp. 280-313 (2016)
26. Gulati, S., Serrà, J., Ishwar, V., Serra, X.: Mining melodic patterns in large audio collections of Indian art music. In: International Conference on Signal Image Technology and Internet Based Systems (SITIS-MIRA), pp. 264–271. Morocco. 9, 87, 124, 148 (2014c)
27. Murthy, H., Bellur, A.: Motif Spotting in an Alapana in Carnatic Music. In: Proceedings of the 14th International Society for Music Information Retrieval Conferences (2013)
28. . Rao, P., Ross, J., Ganguli, K., Pandit, V., Ishwar, V., Bellur, A., & Murthy, H.: Classification of Melodic Motifs in Raga Music with Time-series Matching. In: Journal of New Music Research, 43, 115 - 131 (2014)
29. Ganguli, K., Gulati, S., Serra, X., Rao, P.: Data-Driven Exploration of Melodic Structure in Hindustani Music. In: Proceedings of the 17th International Society for Music Information Retrieval Conference (2016)
30. S. Gulati, J. Serra, K. K. Ganguli, X. Serra.: Landmark detection in hindustani music melodies. In: International Computer Music Conference Proceeedings (2014)
31. Hennequin, R., Khlif, A., Voituret, F., Moussallam, M.: Spleeter: A fast and state-of-the-art music source separation tool with pre-trained models (2019)
32. Srinivasamurthy, A., & Gulati, S., & Caro Repetto, R., & Serra, X.: Saraga: Open dataset for research on Indian Art Music. Empirical Musicology Review. `https://compmusic.upf.edu/` [Preprint] (2020)
33. Fuentes, M. et al.: mirdata v.0.3.0. Zenodo. `http://doi.org/10.5281/zenodo.4355859` (2021)
34. Gulati, S., Serrà, J., Serra, X.: Improving Melodic Similarity in Indian Art Music Using Culture-Specific Melodic Characteristics. In: Proceedings of the 16th International Society for Music Information Retrieval Conference (2015)
35. Bhagyalekshmy, S.: Ragas in Carnatic Music. Trivandrum: CBHH Publications (1990)
36. Gulati, S., Serrà, J., and Ganguli, K., Sertan, Ş., and Serra, X.: Time-delayed melody surfaces for Raga recognition. In: Proceedings of the 17th International Society for Music Information Retrieval Conference, pp. 751–757 (2016)

Noise Reduction Using Self-Attention Deep Neural Networks

Naoyuki Shiba and Hiroaki Saito

Graduate School of Scienece and Technology, Keio University, Japan

Abstract. In recent years, there has been a lot of research on the task of source separation, which is the separation of sound sources from a piece of music into vocal and accompaniment components. This paper proposes a model that introduces self-attention to Open-Unmix, an open source software for the source separation task. Self-attention is a mechanism that learns and determines the data flow itself in neural networks. We applied this model to a speech separation task to remove noise from speech, and compared it with previous methods using an objective evaluation measures (SDR, ISR, SAR). The results show that a proposed method outperform the previous methods in SDR. Furthermore, the t-test showed a significant difference between the two methods.

1 Introduction

Recently developed deep learning techniques have been used in the study of sound source separation, and their accuracy has been dramatically improved. Source separation refers to the task of extracting a single source from a mixture of sources. An example is to separate the signals of specific instruments from a pop music piece. This technique is useful for removing vocal components to create a karaoke sound source, or for creating a score for each instrument in a piece of music. Research on music source separation has been actively conducted to expand the number and types of sources to be separated as well as the accuracy of the separation. Another source separation task is to remove noise from a noisy speaker's speech signal. This paper proposes a method for extracting speech from noisy speech by removing noise.

A neural network propagates data from input to output in a computational manner according to a pre-designed network structure. For many problems, the performance can be improved by designing the structure using prior knowledge. However, it is difficult to improve the efficiency of learning in areas that cannot be compensated by prior knowledge. Self-attention is a method designed to deal with these problems by learning and determining the way the data flows itself, paying attention to the results of its own intermediate calculations, and calculating relevance by paying attention to all positions in the same sequence. It has been applied in fields such as machine translation and image generation [6][8]. Self-attention has also been applied to source separation [3], where each time segment is associated with other time segments that share the same repetitive patterns, and these repetitive patterns are used as additional information for source separation. Self-attention is an attention mechanism that indicates the similarity and importance between the elements of itself, and for each element it calculates the

query Q, key K, and value V using that element (the same values are used for Q, K, and V). When d_k is the dimensionality of the query and key, it is computed as in Eq. (1). The inner product of the query and the key is calculated and divided by the number of dimensions to take into account the context of the whole series, and then the softmax function is applied to prevent the gradient from being lost. Then, by multiplying the calculated weights by the same values as the original input, the output takes into account the context of the training.

$$\mathrm{attention}(Q, K, V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \tag{1}$$

Self-attention controls which elements are weighted based on the similarity of the query and key. As a result of learning, if it is better to read from an element, the corresponding query and key are updated to be closer, and if it is better not to read from an element, they are updated to be farther apart. In this way, the system automatically decides which element to read.

In [3], self-attention was introduced into a model called Dense-Unet, and it was shown that self-attention leads to improved accuracy in source separation. Dense-Unet was created by synthesizing a structure called Dense Net, which directly connects all layers, and a structure called U-Net, which has skip connections to pass information at each layer between encoders and decoders. Although the Dense-Unet model showed high accuracy compared to previous studies, they showed that the accuracy was further improved by introducing self-attention.

In addition to the above studies, various other methods have been proposed for source separation models. In a study of source separation using a transfer learning approach [5], a model used for speech recognition is trained on a large dataset, and the features are transferred to a source separation model using DenseNet, thereby solving the conventional problem of not being able to maintain long-term dependencies. In this paper, we present an audio query-based separation. In a study of audio query-based separation [2], various types of sound sources are separated by directly compressing the same sound source as the musical instrument to be separated into a latent vector and feeding it to the U-Net model as the target information to be separated.

The purpose of this study is to further improve the accuracy of Open-Unmix, a high-performance and open-source source separation model, by introducing self-attention. Self-Attention is introduced to improve the performance of separation by exploiting long-term internal dependencies when the noise is repeated. Open-Unmix[4] is a three-layer bi-directional LSTM model that takes the spectrogram of the mixed sound source as input and learns to predict the spectrogram of the target sound source for each instrument and vocal of the song. The model learns a mask to remove all sources except the target source, and performs source separation by multiplying the input source by the mask. Among open-source sound source separation software, it shows very high accuracy in the separation results. In addition, we apply the model used for sound source separation of music to the task of sound separation, which is to remove noise from noisy speech information[1], showing that sound source separation research can be applied to various tasks.

[1] https://github.com/seth814/open-unmix-pytorch

2 System Description

In this section, we describe the source separation system proposed in this paper.

2.1 Input Data

In this study, we created a database consisting of speech and noise that can be adapted to the input format of Open-Unmix, a source separation model for music. A mixture of speech and noise was created, and the system was trained to separate them. For noise data, we used the ESC-50[2] dataset. ESC-50 is a dataset of 50 classes and 2,000 files of environmental sounds. ESC-50 is a dataset of 2,000 files of 50 classes of environmental sounds, including animal sounds, rain sounds, human coughs, clock alarms, engine sounds, and other sounds without voices (words). Each file is 5 seconds long and has a sampling rate of 44.1 kHz. It consists of audio data (.wav) and metadata (.csv), and the metadata contains the file name, class (0-49), and class name.

For the audio data, we used the publicly available podcast[3] data. We used the podcast data because the speakers speak clearly and there is little noise. We used the data of 10 broadcasts (95820 seconds).

The data set was pre-processed. First, since each sound source of ESC-50 is 5 seconds long, we converted the podcast audio data into a wav file and divided it into 5-second segments. Then, for each sound source, the data was divided into 80 % for training data, 10 % for validation data, and 10 % for test data. Open-Unmix supports data input in the form of source folders rather than track folders, and the data loader loads random combinations of target and interferograms as input. The model then estimates the mask of the target, and finally outputs the target.

2.2 Proposed Model

The proposed method consists of Open-Unmix and self-attention. These are described by Pytorch[4]. The model structure is as follows (Fig. 1).

The input signal is first converted into a spectrogram by STFT. The input spectrogram is standardized using the mean and standard deviation of each frequency bin over all frames. In addition, batch normalization is applied at several points in the model to stabilize the training. When training with LSTM, the frequency and channel axes of the input information are compressed before training, instead of using the original input spectrogram resolution. This is expected to reduce redundancy and training time. Open-Unmix is composed of three layers of BLSTM. After applying BLSTM, the signal is decoded and returned to its original input dimension. The output is finally multiplied by the mixture of sources as a mask is generated to separate the target sources. In order to perform the separation to multiple sources, the model is trained simultaneously for

[2] https://github.com/karolpiczak/ESC-50
[3] http://podcasts.joerogan.net
[4] https://pytorch.org/
[5] https://github.com/sigsep/open-unmix-pytorch

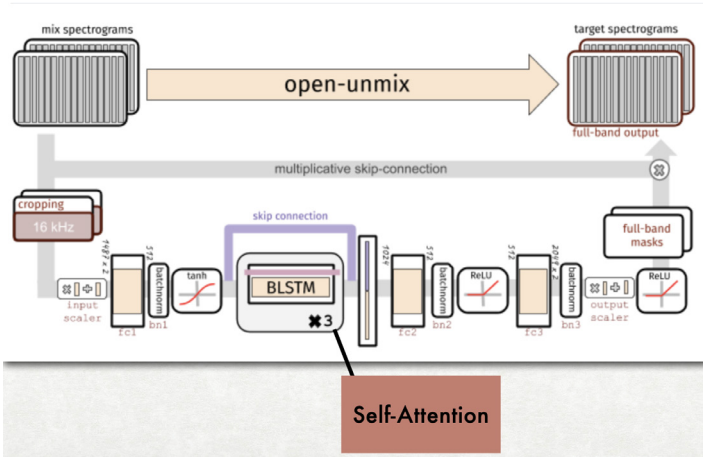


Fig. 1. Diagram of the Open-Unmix[5] with self-attention.

each specific target. In the case of this study, a model for noise extraction and a model for speech extraction are trained simultaneously.

As shown in Fig. 1, we combine self-attention with the output of BLSTM to weight which elements should be focused on during training. The input size to BLSTM is (255,128,512), and the output is returned as a tuple, passing the first element, the hidden layer vector. We concatenate the output of BLSTM and the information held by the skip connection, and the size becomes (255,128,1024). Then, the weighting by self-attention is added.

3 Evaluation Results And Comparisons

3.1 Experiment

We compare the speech separation accuracy of the proposed method with that of Open-Unmix alone. For self-attention, we set the number of channels to be compressed in the convolutional layer to 100 and the output size in the linear layer to 32. The other parameters are batch size 128, window length 512 for STFT, hop count 160 for STFT samples, and data format .wav. The other parameters in Open-Unmix are set by default. SDR, ISR, and SAR were used as evaluation indices [1]. These indices were calculated using the museval package[6]. The units are all expressed in dB, and the larger the value, the higher the accuracy.

3.2 Results

Table 1 shows the results of the objective indices for the test data of the previous and proposed methods. We evaluated the results for both the separated voice and noise.

[6] https://github.com/sigsep/sigsep-mus-eval

The following functions are used to measure performance: Source to Distortion Ratio (SDR), Image to Spatial distortion (ISR), and Sources to Artifacts Ratio (SAR) [1][7].

Table 1. Comparison of previous and proposed methods. Evaluations were calculated for speech and noise respectively.

Metric(dB)	Open-Unmix	Proposed method
SDR (voice)	6.83	7.46
SDR (noise)	8.43	8.59
ISR (voice)	10.24	8.97
ISR (noise)	12.55	12.86
SAR (voice)	6.38	6.53
SAR (noise)	8.92	9.02

All the noises in ESC-50 and the podcast were synthesized, and the evaluation index of speech separation was calculated for each noise and represented using a box-and-whisker diagram.

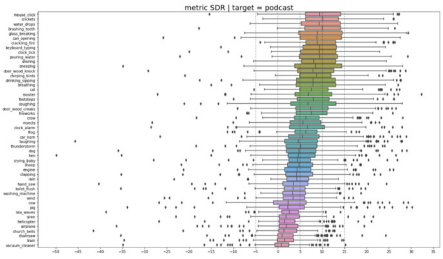


Fig. 2. SDR for all noises (Open-Unmix).

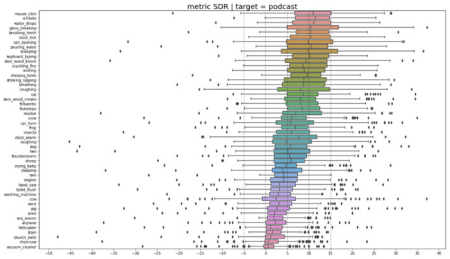


Fig. 3. SDR for all noises (Proposed method).

3.3 Discussion

A t-test was conducted to determine if there was a significant difference in the evaluation of the results. According to Table 2, there was a significant difference in the improvement of accuracy for both voice and noise in SDR, and a significant difference in the improvement of accuracy for voice and noise in SAR. On the other hand, ISR showed a decrease in accuracy in voice.

Since SDR is an overall evaluation value that includes all other evaluation metrics, the improvement in SDR indicates that self-attention is useful for voice separation. On the other hand, the accuracy of the ISR for voice has decreased. It is possible that the weighting of self-attention was not done correctly, or that there was a problem in the model.

Table 2. t-test for evaluation measures.

	SDR(voice)	SDR(noise)	ISR(voice)	ISR(noise)	SAR(voice)	SAR(noise)
t-ratio	-0.62	-1.89	10.35	0.65	1.33	0.1
degree of freedom	199	199	199	199	199	199
significance level	0.05	0.05	0.05	0.05	0.05	0.05
t-distribution	1.65	1.65	1.65	1.65	1.65	1.65
significance of test	P < 0.05	P < 0.05	P > 0.05	P < 0.05	P < 0.05	P < 0.05

The value of SDR changes depending on the type of noise (Figs. 2, 3). This is due to the fact that intermittent noises and noises that are not too loud tend to have higher accuracy than continuous noises during 5 seconds. While none of the previous methods exceed 35 dB, the proposed method exceeds it for three noises.

4 Conclusion

In this paper, we attempted to further improve the accuracy of the Open-Unmix model, which has high performance in open source software, by introducing self-attention. In addition, we demonstrated the versatility of this model for the source separation task by using it not for the source separation task, which separates vocals and accompaniment from a piece of music, but for the speech separation task, which separates each from a mixture of voice and noise. The results of the t-test showed a significant difference.

References

1. Fevotte, C., Gribonval, R., Vincent, E.: BSS EVAL toolbox user guide Revision2.0, Technical report, IRISA (2005)
2. Lee, J. H., Choi, H.-S., and Lee, K.: Audio query-based music source separation, in Proceedings of the International Society for Music Information Retrieval Conference 2019, p. 878–885 (2019)
3. Liu, Y., Thoshkahna, B., Milani, A., and Kristjansson, T.: Voice and ac- companiment separation in music using self-attention convolutional neural network, in arXiv:2003.08954 (2020)
4. Stoter, F.-R., Uhlich, S., Liutkus, A., and Mitsufuji, Y.: Open-Unmix - a reference implementation for music source separation, in Journal of Open Source Software (2019)
5. Takahashi, N., Singh, M. K., Basak, S., Sudarsanam, P., Ganapathy, S., and Mitsufuji, Y.: Improving voice separation by incorporating end-to- end speech recognition, in Proceedings of IEEE ICASSP 2020, pp. 41–45 (2020)
6. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I.: Attention Is All You Need, in Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
7. Vincent, E., Sawada, H., Bofill, P., Makino, S., and Rosca, J.: First stereo audio source separation evalution campaign: data, algorithms and results, in Proceedings of ICA 2007, pp. 552–559 (2007)
8. Zhang, H., Goodfellow, I., Metaxas, D., and Odena, A.: Self-Attention Generative Adversarial Networks, in International Conference on Learning Representations (2019)

Estimation of Perceptual Qualities of Percussive Sounds Inspired by Schaefferian Criteria: Attack Profile, Mass, and Harmonic Timbre

Sérgio Freire, José Henrique Padovani and Caio Campos

Federal University of Minas Gerais
sfreire@musica.ufmg.br
jhp@ufmg.br
costacaiocampos@gmail.com

Abstract. Pierre Schaeffer's typomorphology (1966) proposes seven criteria of musical perception for the qualification of sound objects under reduced listening; these criteria form the basis of a theory (solfège) of musical objects fitted to musical contexts where pitch is not the most relevant feature. We developed a real-time setup that uses low-level audio descriptors to identify and classify percussive sounds as bundles of features related to Schaefferian concepts. The paper describes a segmentation method and the tools and strategies used for addressing three of these criteria: attack profiles (as genres of the criterion dynamic) and mass (which closely relates to the criterion harmonic timbre). The examples depict quantitative results and discuss their correlation with perceptual qualities.

Keywords: typomorphology; Pierre Schaeffer; percussion; audio descriptors

1 Introduction

In recent years, many papers have sought to bring the tools and procedures related to audio descriptors and Music Information Retrieval closer to the theoretical and methodological contributions of Pierre Schaeffer. Such works seek to correlate Schaefferian morphological descriptions and the quantitative data extracted through various techniques of digital signal processing, aiming to automatically index sounds [1], associate "subjective labels" and acoustic features [2, 3], or conjugate descriptors data and perceptual criteria in analytical contexts by employing statistical methods [4], among other approaches [5, 6].

In this paper, we aim to approximate the Schaefferian *solfège* criteria and low-level audio descriptors by implementing real-time audio analysis processes using the computer music language/environment *Max*[1]. In this study, we have chosen the universe of percussive sounds since their sonic qualities are represented and qualified in a very rough manner when using the concepts and parameters of traditional music theory (notes and durations). Bringing into account concepts like complex sound, mass profile, grain, among others, we hope to develop a more efficient tool for the qualification of these sounds to be used in interactive contexts. In this article, we analyze a limited but

[1] https://cycling74.com/products/max

varied selection of percussive sounds, and the results show the suitability of the chosen descriptors for the qualification and differentiation tasks.

The text is organized as follows. Firstly we summarize the seven criteria of musical perception as defined by Schaeffer. After that comes the description of the dataset and procedures used in audio pre-processing and segmentation. A central section deals with the chosen audio descriptors and their correlations with Schaefferian concepts. A set of selected examples comes next. Finally, we present the next steps of the project.

2 Schaeffer's Criteria of Musical Perception

In the sixth book of his *Traité des Objets Musicaux* (TOM), Pierre Schaeffer presents one of the most remarkable contributions of his research: a proposal of a generalized *solfège*, dedicated not only to the traditional musical notes, but also to any sound considered "potentially musical". This *solfège* entails seven typomorphological criteria of musical perception that have the purpose to guide the listening process that consciously attempts to detach the sonic characteristics from any referential or causal events that may generate sound objects themselves: a method that Schaeffer, borrowing the Husserlian concept of *epoché*, named *reduced listening*. In section 34.3, the TOM includes a Summary Diagram (*tableau général*), offering an overview of the whole method, where types, classes, genres, and species of sound objects are described according to seven criteria — mass, dynamic, harmonic timbre, melodic profile, mass profile, grain, and *allure* [7, 8][2]. The criteria help to locate the position and thickness (*site*/*calibre*) of sound object attributes in the three-dimensional space of a perceptual field formed by *pitches*, *durations*, and *intensities*.

In section 88 of the *Guide des Objets Sonores* [9], Michel Chion outlines the distinctive features of sound objects that each of the morphological criteria proposed by Schaeffer aims to evaluate.

- The *mass* details how the sound occupies the pitch perceptive dimension.
- *Harmonic timbre* describes the "diffuse halos" and "related qualities" that seem to be related to the *mass* and allow its qualification.
- *Grain*, in its turn, is related to the "micro-structure" of sound matter and is associated with rapid variations or reiterations of constituent sounds.
- While *grain* outlines the link between form and matter as one of the sustainment criteria, *allure* expresses the dynamism (mechanical, living, or natural) of what could be defined as a "generalized type of vibrato".
- *Dynamic* expresses the evolution of a sound in the perceptive dimension of *intensities*.
- *Melodic profile* describes the general contour of a sound in the perceptive dimension of pitches, a sort of trajectory in the tessitura.
- *Mass profile*, on the other hand, describes the "internal" variations of a sound in this same perceptive dimension: these changing shapes are responsible for "sculpting" the mass, making it to be more or less thick or thin, having thus a more or less complex or tonic quality, for instance [9].

[2] See, particularly, pp. 584-587 of the original edition; pp. 464-467 of the English translation.

As a general method, *reduced listening* involves a conscious attitude that refrains the habitual curiosity towards the sound sources and their meanings in favor of addressing intrinsic features of the sonic phenomena. While authors like Di Scipio [10] remark that the concept of *reduced listening* is ideologically and technologically circumscribed, ignoring the very audible traces of electroacoustic tools that enable us to focus on the 'sound itself', its relevance, since the Schaefferian seminal contributions, lies in the fact that the project of a "generalized *solfège*" has been successful in providing a rich theoretical framework that makes possible to describe different features, behaviors, and qualities of sound objects according to morphological criteria and perceptual dimensions.

Considering Pierre Schaeffer's well-founded warnings regarding the differences between the study of sound objects using perceptual-sensory criteria, on the one hand, and physical-acoustic analysis of audio signals, on the other, it is relevant to underline the experimental nature of the present work. Thus, despite the differences between perceptual and signal-based evaluation, description, and categorization of sounds, our work is motivated by a common trait of low-level audio descriptors and the Schaefferian *solfège*: both focus on intrinsic qualities of sound phenomena, seeking to discriminate particular characteristics based on certain criteria, dimensions, or parameters. Indeed, Schaeffer himself, even warning to the differences between perceptual processes and what signals can represent, also recognized the usefulness of real-time visualization of signals using bathygraphs and sonagraphs. [7, 8] [3].

3 Selection of Sounds, Pre-processing and Segmentation

In our program, the expected inputs are audio streams delivered by microphones, pick-ups, or mixers, featuring different background noise levels and dynamic ranges. In the current phase, we have chosen to use a set of pre-recorded sounds. This procedure offers not only variety but also repeatability, two relevant factors for building and improving tools. The sound selection, depicted in Table 1, was based on Schaefferian types. These recorded sounds function as live inputs to the setup[4], which runs with a sampling frequency of 48 kHz.

The sounds are segmented between onset and offset points. In some situations, a new segmentation clue may occur before the offset; in these cases, this clue determines the offset of the previous event and the beginning of the current one, characterized as "slurred". The detection of onsets and offsets occurs by comparing an RMS envelope (expressed in dBFS) with two thresholds, 6 dB and 3 dB, respectively, above the background noise level. This envelope uses a very short window for its estimation — 256

[3] In the pp. 556-557 of the English translation of the TOM [8]: "It is perhaps disconcerting to see us, after so many warnings, recommending the use of the bathygraph and the Sonagraph to describe a piece of music.(...) On the physical level the bathygraph and the Sonagraph give two graphs of the signal in real time: its projection on the dynamic and the harmonic plane. Of course, these lines are not very intelligible because perceptions of sound differ so much (by anamorphosis) from the signal on the printout."

[4] The soundfiles used in this study are available in the following repository: `https://github.com/lapis-ufmg/2021_CMMR_arquivos`

Table 1. Selected sounds.

sound	description	sound	description
tabla.gliss	single tabla stroke with glissando	rattle	single directional rattle shake
tomtom	single tomtom stroke	tamb.tremolo	tambourine tremolo
whip	single whip attack	berimb.jete	berimabau jete, multiple strokes
tamb.slap	single tambourine hand slap	berimb.vib	single berimbau stroke, with vibrato
sdrum.drag	snare drum drag, with snare	pand.rim.frict	pandeiro tremolo-like rim friction
sdrum.nosnare	single snare drum stroke, without snare	sleighbells	multiple sleighbells shakes
bassdrum	single bassdrum stroke	thunder.shake	multiple thunder sheet shakes
cymbal	single cymbal stroke	rainstick	rainstick tip
gong.tuned	single tuned gong stroke	timp.roll	timpani roll
gong.untuned	single untuned gong stroke	whistle	single whistle blow
guiro	single directional guiro rub	pand.skin.frict	single pandeiro skin friction
ratchet	single ratchet swing	vibes.bow	single vibraphone key bow
cymbal.bow	single cymbal bow		

points and hop size of 64 —, which will be referred to as *rms256:4*. In order to obtain more efficiency and precision, we implemented this curve with the Max `[gen~]` object, which uses native audio signal processing routines. Its output is smoothed with a low-pass filter (a single one pole filter, with a –6 dB per octave attenuation), and different cutoff frequencies are employed, depending on the purposes of its use. We use a cutoff frequency of 4 Hz in the estimation of onsets, offsets, and attacks. In the latter case, the audio stream may pass through a filter before the calculations. The estimation of attack profiles and iterative grains uses this same signal with a cutoff frequency of 30 Hz. A control-rate version of this envelope builds the attack profile. Onsets and offsets also function as gates for other processing tools in time and frequency domains. These processes explore the data delivered by a *rms2048:4* curve and by spectral peaks values estimated by the `[sigmund~]` object [11], using the same window and hop size.

4 Perceptual Attributes and Algorithms

Due to text size restrictions, we will concentrate on a subset of Schaefferian perceptual criteria (or sub-criteria) and their correlated audio tools, namely the *attack profile*, *mass*, and *harmonic timbre*.

4.1 Attack Profile

The importance of the attack portion of sounds has been stressed clearly in the work of Pierre Schaeffer, deserving special attention in the TOM and *Solfège*. At that time, he complained that physical measurements were far from representing an accurate picture of the perceived sonic dimensions. About the initial transient portion of sounds (ca. 50 ms), he observed that: “A more spectacular experiment involved asking a very good

trumpet player to play a staccato with an accuracy appreciable to the ear: none of this sound's eight impulses gave an oscillogram similar to the others (fig. 5)" [8], (p. 164).

It is not our purpose to argue about how different oscillograms associate with similar perceptions of the beginning of sounds. On the other hand, we intend to approach the initial portion of sounds (ca. 400 ms) with a descriptor that allows for an association between audio features and perceived qualities, using a tool similar to the "bathygraphic traces" depicted by Schaeffer [7, 8][5]. This approach is similar but not equal to the estimation of log-attack-times and attack slopes [12]. We prefer to analyze the entire profile, which may surpass 300 ms, instead of stopping at the point usually named the end of attack. For the same reason, this point will be called the *attack first plateau*. The difference between the levels of the first plateau and the onset is named *attack size*, and the time interval between them *attack duration*. The slope of the first plateau (**FPSlope**) is the ratio between these two values.

The first plateau is estimated as the instant when the derivative of the low-pass filtered audio-rate *rms256:4* curve from the (possibly filtered) input audio stream crosses (or comes near to) zero, just after having surpassed a predetermined positive threshold (a *sharpness* parameter). A value of 200 ms is set as the default reattack threshold since we prefer to consider multiple fast strokes (such as flans, drags, ricochets) as belonging to the same profile. Depending on the settings (filtering and thresholds), this estimation may not produce results for soft attacks[6]. We also prefer to consider some iterative sustainments as a single object displaying *allures*, even when the distance between peaks exceeds the reattack threshold. The onset of a slurred event also marks the offset of the previous one and is defined as the instant when the already mentioned derivative surpasses the given threshold.

The attack profiles depend on their context of production, mainly on the dynamic level and duration (which is also related to excitation and sustainment types). Schaeffer defines seven genres: *abrupt*, *steep*, *soft*, *flat*, *gentle*, *sforzando*, and *nil*. The first three genres relate to different attack-resonance types; a sudden burst of energy characterizes the flat profile; the *gentle* genre has no apparent attack; *sforzando* generally associates with short sounds with a characteristic crescendo; *nil* points to the very progressive emergency of a profile. For the sake of comparison, we have kept all the parameters, except the background noise threshold, fixed for all sounds selected for the present study. The first 300 points after the onset, corresponding to 400 ms, are plotted on a user screen, and stored in a buffer, for further analysis. Figure 1 depicts the attack profiles of nine sounds.

4.2 Mass / Harmonic Timbre

Schaeffer defines seven classes of mass: pure sound, *tonic*, *tonic group*, *channeled*, *nodal group*, *node*, *white noise*. Pure and tonic sounds present a clear pitch, while tonic

[5] See p. 533 of the original edition; p. 425 of the English translation.

[6] These parameters (reattack time and sharpness) help to redefine the fluids limits between the *context* ("whether the criteria are artificially put into a structure...") and the *contexture* ("...or naturally form a structure") of percussive sounds in contexts with different segmentation clues. The quotations are from p. 402 of the English translation [8].

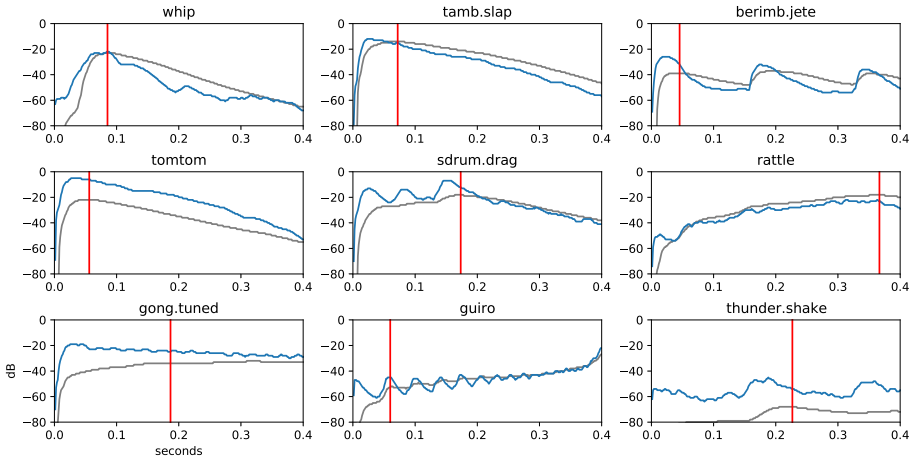


Fig. 1. Attack profiles and first plateau (red mark) of nine percussive sounds. The gray curve is a *rms256:4* low-pass filtered (4 Hz) envelope of the low-pass filtered input signal, and is used for the estimation of the plateau. The blue curve is a *rms256:4* low-pass filtered (30 Hz) envelope of the non-filtered input signal, and is used for the qualification of the attack profile. Filters and threshold parameters remained unchanged for all sounds.

groups are chord-like sounds. Node is a filtered noise occupying a definite spectral region, and nodal group a combination of nodes. Channeled sounds are an ambiguous class between the pitched and unpitched sounds. Within percussion instruments, the most common classes are tonic, channeled, node, and nodal group. However, the latter occurs more like a combination of single nodes than as a single sound object. Harmonic timbre is a complementary criterion to mass, being almost inseparable in some situations. In Schaeffer's words: "Consequently we intend to use the two criteria of mass and harmonic timbre in conjunction with each other, considering them rather as connecting vessels, with the exception of certain specific examples(...)" [8][7]. These specific examples are the pure and tonic sounds. Therefore, we have opted to use the same audio descriptors for both criteria. Typical harmonic timbre attributes are expressed by terms like *full*/*hollow*/*narrow*, *rich*/*poor*, and *bright*/*matt*. The last attribute pair may overlap with the judgment of the spectral region occupied by the mass of a given sound.

Our strategy to deal with the main classes of mass, and the associated harmonic timbres in the percussive realm, relies on the analysis of spectral peaks and (monophonic) pitches estimated by the `[sigmund~]` object. As we are presently not interested in isolating individual notes inside chords, these two outputs are sufficient for our purposes. Since our context does not presuppose the existence of a harmonic series of spectral peaks, we can not use traditional descriptors as noisiness, inharmonicity, odd-to-even harmonic ratio, tristimulus, among others.

The following descriptors use the spectral peaks (up to 20) estimated for each analysis frame and the energy equivalence given by Parseval's theorem. Their output is a

[7] Quoted from p. 412.

curve with a refresh rate of 10.67 ms. Percentile 50 (**pct50**) and percentile 80 (**pct80**) are the number of spectral peaks needed to obtain 50% (-3dB) and 80% (-1dB) of the total energy present in the signal. For sounds with a broad spectral distribution, 20 peaks may not reach the chosen percentiles; in these cases, the output will be 20 peaks, and further information is to be delivered by the next descriptor. The percentage of sound energy represented by up to 20 peaks (**20P/total**) is calculated for each frame and expressed in values between 0 and 1. The frequency of the most prominent peak (**MPP**) in each frame is expressed in Midicents[8]. The object `[sigmund~]` outputs a value in Midicents for the pitched frames, and the value -1500 for unpitched frames. Our descriptor outputs a scalar (percentage of unpitched to total frames: **unpitched/total**) and a curve with all values. In this curve, the unpitched values are represented by the number 1. The estimation of the intrinsic **dissonance** uses the algorithm developed by Sethares [13].

The spectral centroid (**SC**), or the center of gravity of a spectrum, is estimated with a `[gen~]` routine delivered with the Max program since its version 6. Instead of using a nominal value in Hz, we use values in Midicents, which define a scale ranging from 15.5 to 155 in the audible range. The difference between the lowest and highest peak frequencies (Δ **peaks**) is also expressed in Midicents. The spectral **region** is estimated from the contribution of each peak to different spectral ranges. The first three octaves (20–160 Hz) define the low range, the four intermediate octaves (160–2,560 Hz) the medium range, and the last three octaves (2,560–20,000 Hz) the high range. If none of these ranges carry 40% or more of the total energy, the sound frame is classified as wideband, labeled as (7). Otherwise, any range with more than 40% of the total energy contributes to qualify one the six spectral combinations: (1) Low, (2) Low/Medium, (3) Medium, (4) Low/High, (5) Medium/High, (6) High.

4.3 Time Series Statistics

Most of the descriptors detailed above are represented by time series, which are subjected to simple statistical analysis just after the offset. Our implementation adapts the algorithms given in [12], and we have chosen the following scalar descriptors: mean value and standard deviation; temporal centroid and spread (normalized by the total duration); skewness; kurtosis; crest; flatness. These values will support the correlations with the Schaefferian perceptual attributes.

5 Examples

For each live input sound, our program generates real-time curves (or markers) for all descriptors and calculates the scalar values described in section 4.3. The results of the analysis of the attack profiles shown in Figure 1 appear in Table 2. We will focus firstly on the sounds produced by one single stroke (percussion-resonance type). Perceptually, the whip sound presents an abrupt profile. This attribute correlates with its short duration, low values for temporal centroid and spread, a positive skewness, and a high crest.

[8] Midicent is the unit of a logarithmic scale for frequencies, in which the value 69 represents 440 Hz (note A4), and each integer step is an equal-tempered semitone.

The tomtom stroke presents a steep profile. Although it has a temporal centroid similar to the whip, its spread is much larger, and the skewness and crest are less pronounced. The profile of the tambourine slap lies in between these two sounds. The tuned gong stroke presents a soft profile; in this case, we have a "reinforcement of the resonator". A longer duration, a slightly positive skewness, and a high value for flatness correspond to this attribute. The rattle profile, produced by a shake, is perceived as gentle since an initial shock is absent. The sound production mixes iterative and continuous aspects. The negative skewness, a small crest, and medium flatness point to this attribute. There are three sounds with a clear iterative or granular profile: the guiro rub, the berimbau jeté, and the drag on a snare drum. The parameters of the two latter are halfway between the steep and the soft profiles; in them, a new stroke prolongs the short resonance. On the other hand, the guiro has a sforzando profile: a high temporal centroid, a considerable negative skewness, a high crest. We believe that the iterative character could integrate the basic profiles as a second-order qualifier. Finally, we have the long thunder shake, whose profile approaches the genre nil: its medium temporal centroid, a low value for crest, and a high value for flatness corroborate this qualification. In this case, the dynamic level (the average value of the entire object) indicates that a marked crescendo will happen during its course. In general, long sounds will rely less on their attack portion for their characterization. It seems to us that the use of the slope of the first plateau (which may vary significantly according to specific parameter settings) and kurtosis could be more significant with more homogenous sounds. In the present selection, they are not as meaningful as the other parameters.

Table 2. Attack parameters for nine selected percussive sounds (the same from Figure 1), plus total duration and dynamic level.

sound	FPSlope (dB/ms)	temp. centroid	temp. spread	skewness	kurtosis	crest	flatness	dur (ms)	DL (dB)
whip	0.42	0.24	0.26	1.25	2.87	6.08	0.30	396	-46
tamb.slap	0.85	0.22	0.74	0.31	0.25	4.07	0.42	461	-33.5
tomtom	1.03	0.24	1.18	0.17	0.09	3.60	0.43	487	-28.5
sdrum.drag	0.33	0.35	1.06	0.08	0.10	4.28	0.61	616	-31.8
guiro	0.63	0.72	0.39	-0.66	1.36	11.09	0.67	631	-42
rattle	0.13	0.67	0.67	-0.21	0.28	2.18	0.67	1151	-42
berimb.jete	0.81	0.35	0.53	0.33	0.66	4.77	0.60	2019	-55.6
gong.tuned	0.24	0.42	1.13	0.08	0.11	1.96	0.92	9219	-42.6
thunder.shake	0.18	0.53	0.20	-0.19	3.82	2.80	0.86	15541	-33

The discussion about mass and harmonic timbre relies on data displayed in Tables 3 and 4. A significant presence of pitched frames points to a tonic sound; the opposite indicates a node or a channeled sound. In addition, the concentration of energy in a few spectral components helps differentiating between tonic and channeled sounds on one side and nodal sounds on the other. The combination of these two descriptors can discriminate between *tonic* (bass drum, whistle, friction of a tambourine skin, and tabla), *channeled* (tomtom, tuned gong, and snare-drum without snares), and *nodal* (cymbal,

guiro, ratched, and rattle, among other) sounds. The stability of the most prominent peak may also favor *tonic* and *channeled* sounds, but one must be careful with the presence of melodic profiles, as in the case of the tabla sample. Although intrinsic dissonance values may also point to tonic sounds, this is not a univocal association since spectral peaks not presenting a simple harmonic ratio only increment this value if they are close enough in frequency (see details in [13]). The interpretation of spectral centroid (and region) values is straightforward; however, a high value for the standard deviation indicates the presence of some profile (glissando, undulation, filtering, etc.). We can analyze the shape of these profiles with tools similar to those used for the attack profiles.

In addition to the symbiotic relation between mass and timbre, Schaeffer states that "there is no classified index for perceptions of harmonic timbre" (TOM, p. 420). In the present case, we try to approximate the perceptive attributes *full*/*hollow*/*narrow* with the descriptors Δ peaks, pct50, pct80, and region (helped by the intrinsic dissonance), and the opposition rich/poor with the values estimated for pct80 and 20P/total. For instance, observing different sounds classified in the medium region, it is possible to split them between *hollow* (bass drum, tomtom) and *narrow* (whistle, rattle) according to these parameters.

Table 3. Mass and harmonic timbre parameters (1) for 10 selected percussive sounds.

sound	**dur**	**pct50**	**pct80**	**20P/total (ratio)**	**unpitched/total (ratio)**
tabla.gliss	227	1.3 ± 0.5	6.6 ± 8.5	0.85 ± 0.12	0.24
tomtom	487	1 ± 0.2	1.96 ± 3.1	0.96 ± 0.1	0.38
sdrum.nosnare	560	1.1 ± 0.4	2.4 ± 3.7	0.96 ± 0.1	0.32
ratchet	753	19.8 ± 0.6	20 ± 0	0.42 ± 0.1	0.86
rattle	1103	8.4 ± 2.2	$19.7 \pm 1.3.$	0.71 ± 0.1	1.0
pand.skin.frict	1915	1 ± 0.2	1.5 ± 2.1	0.98 ± 0	0.07
tamb.tremolo	1365	12.5 ± 4.2	20 ± 0	0.62 ± 0.1	0.77
whistle	1463	1.4 ± 0.6	5.2 ± 6	0.89 ± 0.1	0.13
bassdrum	3671	1.1 ± 1	1.2 ± 1.5	0.98 ± 0.1	0.04
gong.tuned	9219	1.3 ± 0.8	2.2 ± 1.6	0.98 ± 0.05	0.42

6 Final remarks

The results obtained so far have demonstrated that our setup can qualify and differentiate diverse types of percussive sounds with a good approximation to the Schaefferian criteria. We believe that we have shown the importance of attack profiles for percussive sounds and the pertinence of the implemented tools for the qualification of mass and harmonic timbre. The next planned steps are the work with performers in real-time interactive contexts (when the pandemic allows), the choice of the most efficient descriptors for each intended perceptive feature, the training of a machine learning algorithm, and the development of interactive musical works.

Table 4. Mass and harmonic timbre parameters (2) for 10 selected percussive sounds.

sound	diss	MPP (mc)	Δ peaks (mc)	SC (mc)	region
tabla.gliss	37 ± 22.5	46.2 ± 10.4	69.9 ± 19.3	54 ± 7.6	1.8 ± 1
tomtom	46.2 ± 17.5	55.8 ± 0.2	64.9 ± 15.3	60.8 ± 10.2	3
sdrum.nosnare	45.3 ± 21.9	61.2 ± 7.7	59.4 ± 15.6	65.5 ± 13.3	3
ratchet	122.5 ± 30.9	97.6 ± 9.5	30.7 ± 4.5	110.2 ± 3.6	6.5 ± 1.3
rattle	138.8 ± 42.6	95.5 ± 1.9	17.5 ± 9.1	101.2 ± 2.7	3
pand.skin.frict	43.2 ± 30.5	48 ± 1.6	70.2 ± 6.7	51.7 ± 6.7	1
tamb.tremolo	235.9 ± 91.7	111.9 ± 23.8	35.5 ± 31.7	118.4 ± 2.7	6 ± 0.6
whistle	132.8 ± 29.5	98.2 ± 0.4	31.9 ± 13.3	98.4 ± 2.4	3
bassdrum	22.6 ± 23	27.8 ± 3.8	99.3 ± 21.6	30.5 ± 8.7	1 ± 0.3
gong.tuned	24.2 ± 17.8	61 ± 2.9	76.4 ± 31	68.2 ± 7.6	3 ± 0.2

Acknowledgments. This work has been supported by the Brazilian research agency CNPq (National Council for Scientific and Technological Development).

References

1. Peeters, G., Deruty, E.: Sound Indexing Using Morphological Description. IEEE Transactions on Audio, Speech, and Language Processing 18(3), 675–687 (Mar 2010)
2. Bernardes, G., Davies, M., Guedes, C.: A Pure Data Spectro-Morphological Analysis Toolkit for Sound-Based Composition. pp. 31–38. Proceedings of the eaw2015, Aveiro (2015)
3. Godøy, R.I.: Perceiving Sound Objects in the Musique Concrète. Frontiers in Psychology 12 (2021), https://www.frontiersin.org/articles/10.3389/fpsyg.2021.672949/full
4. Martins, F., Padovani, J.H.: Analysis Informed by Audio Features Associated With Statistical Methods: a Case Study on 'Imago' (2002) by Trevor Wishart. In: Proceedings of the 5º Encontro Internacional de Teoria e Análise Musical. vol. 1, pp. 219–234. Campinas (2020)
5. Ricard, J.: Towards Computational Morphological Description of Sound. Doctorate, Universitat Pompeu Fabra, Barcelona (Sep 2004)
6. Valle, A.: Schaeffer Reconsidered: a Typological Space and its Analytical Applications. Analitica 8(1), 1–15 (2015)
7. Schaeffer, P.: Traité des Objets Musicaux. Éditions du Seuil, Paris (1966)
8. Schaeffer, P.: Treatise on Musical Objects: Essays Across Disciplines. No. 20 in California studies in 20th-century music, University of California Press, Oakland, California (2017)
9. Chion, M.: Guide des Objets Sonores. Buchet/Chastel, Paris (1983)
10. Di Scipio, A.: The Politics of Sound and the Biopolitics of Music: Weaving Together Sound-Making, Irreducible Listening, and the Physical and Cultural Environment. Organised Sound 20(3), 278–289 (Dec 2015)
11. Puckette, M.S., Apel, T., Zicarelli, D.D.: Real-time Audio Analysis Tools for Pd and MSP. In: Proceedings of the International Computer Music Conference. pp. 109–112. San Francisco (1998)
12. Peeters, G., Giordano, B.L., Susini, P., Misdariis, N., McAdams, S.: The Timbre Toolbox: Extracting Audio Descriptors from Musical Signals. The Journal of the Acoustical Society of America 130(5), 2902–2916 (Nov 2011)
13. Sethares, W.A.: Tuning, Timbre, Spectrum, Scale. Springer, London (2005), http://public.eblib.com/EBLPublic/PublicView.do?ptiID=303730

A psychoacoustic-based methodology for sound mass music analysis

Micael Antunes[1⋆], Guilherme Feulo do Espirito Santo[2], Jônatas Manzolli[1⋆], and Marcelo Queiroz[2⋆]

[1] NICS - Interdisciplinary Nucleus for Sound Studies, University of Campinas
[2] Computer Science Department, University of São Paulo
micael.antunes@nics.unicamp.br

Abstract. A *sound mass* is a specific state of the musical texture corresponding to a large number of sound events concentrated within a short time and/or frequency interval. Conceptually, it is associated with the work of György Ligeti, Krzysztof Penderecki, and Iannis Xenakis, among others. Recent studies have investigated *sound masses* via perceptual models, such as *Gestalt* models of perception and *auditory scene analysis*, and also from a more acoustic and psychoacoustic perspective obtained through audio recordings. The main goal of this paper is to propose a methodology for the musical analysis of *sound mass* music through audio recordings. We apply this method in the analysis of a performance of the first movement of Ligeti's *Ten Pieces for Wind Quintet* (1968), and explore relationships between the obtained audio descriptors and Ligeti's concepts of *timbre of movement* and *permeability*, in order to reveal Ligeti's strategies when dealing with musical texture and *sound masses*.

Keywords: sound mass music, musical analysis, audio descriptors, psychoacoustics

1 Introduction

This paper introduces a computer aided musical analysis methodology anchored on audio descriptors. Specifically, psychoacoustic models are applied to study *sound mass* composition. *Sound mass* composition emerges in the context of discussions about perception and 20th century serial music [19]. Noticeably, these discussions were part of the *Darmstädter Ferienkurse*, where composers attended classes and lectures on psychoacoustics, phonetics, information theory, and sound synthesis [5]. Some well-known examples of *sound mass* compositions are the large number of attacks in Ligeti's *Continuum* (1968), the micropolyphony and cluster techniques in his *Chamber Concerto* (1961), and the mass created by *glissandi* and extended techniques of the string orchestra in Xenakis' *Aroura* (1971).

The central idea in *sound mass* composition is to emphasize perceptual features of sound, by exploring the continuum of time and frequency domains to produce sound

⋆ Micael Antunes is supported by FAPESP Grant 2019/09734-3. Jônatas Manzolli is supported by CNPq Grant 304431/2018-4 and 429620/2018-7. Marcelo Queiroz is supported by CNPq Grant 307389/2019-7.

textures with a high level of fusion and inner movement. Perception of *sound masses* is often linked with the limits of sound integration by the ear [19] and *microtime* perception [3]. *Sound mass* music is also associated with Huron's perceptual principles of *minimum masking, pitch proximity and limited density*, which are anchored in the *critical bandwidth* psychoacoustic model [17, 26].

Previous works have studied Ligeti's *sound mass* composition from a perceptual perspective, mainly through the *symbolic analysis* of the score. Clendinning explored such a perceptual approach in the study of Ligeti's compositional techniques such as *pattern-meccanico* [10] and *micropolyphony* [9]. Cambouropoulos [8] used *Gestalt* theory to investigate links between Ligeti's techniques and their perceptual outcomes, an approach already explored by Ferraz [14]. More recently, Douglas et al. [12] investigated *Continuum* (1968) within the context of Bregman's *auditory scene analysis* [6].

Methodologies anchored in audio descriptors with a psychoacoustic approach, which emerged in the context of computational and systematic musicology [23, 36, 21], have also been used to study Ligeti's works [21, 2, 1]. In this paper, we propose a methodology for musical analysis [36] focused on perceptual concepts that motivate *sound mass* music composition, associating them with descriptors derived from audio recordings. Specifically, we study Ligeti's viewpoint on *sound mass* composition through the concept of *timbre of movement* [19], associating it with *loudness* [13] and *roughness* [31]. Due to the correlation between spectral information and the perception of pitches and individual voices [17], we also investigate the use of *spectral entropy* [25] and *spectral irregularity* [7], associating them with Ligeti's concept of *permeability* [19, 2, 1, 3]. We present an musical analysis of the first movement of Ligeti's *Ten Pieces for Wind Quintet* (1968), using score-based information alongside the audio signal of a particular performance of this piece. We also derive representations based on audio descriptors that allow us to discuss Ligeti's compositional strategies and their perceptual aspects, as well as the formal development of the piece from the viewpoints of *timbre of movement* and *permeability*.

In Section 2, we lay out the theoretical background for this study, starting with an exposition of the concepts of *timbre of movement* and *permeability*. Then, we give an overview of the first movement of the *Ten Pieces for Wind Quintet*, followed by a review of the audio descriptors used in this work. In Section 3, we outline the analytical methodology proposed, and in Section 4 we present and discuss the results of our study. Finally, in Section 5, we present our conclusions.

2 Theoretical background

2.1 Ligeti's concepts of *Timbre of Movement* and *Permeability*

Two relevant György Ligeti's concepts associated with *sound mass* music composition are *timbre of movement* and *permeability*.

The concept of *timbre of movement*[3] refers to the achievement of fusion in musical texture by mixing a large number of sound events [19, p. 169]. Ligeti associates this

[3] In the original, Ligeti uses *timbre du mouvement* in French and *Bewegungsfarbe* in German [19, p. 169].

concept with his collaboration with Gottfried Michael Koenig in the electronic studio of the Westdeutscher Rundfunk (WDR) in Cologne [19]. To him, the most meaningful knowledge acquired in the studio was the observation that sound samples or synthesis components merge into a single texture when the number of sounds surpasses a certain threshold of our perception. This occurs when our auditory system can no longer discern the individual components of a musical texture, leading our attention to the global features and inner movements of *sound masses* [19, p. 169]. Ligeti used this concept of *timbre of movement* in his instrumental compositions with the *micropoliphony* technique [9], which allows achieving dense textures by overlapping a large number of melodies with short notes.

The concept of *permeability* refers to a state in which we are unable to distinguish pitches and individual voices. According to Ligeti: *"The loss of sensitivity to intervals is at the source of a state that could be called permeability"* [19, p. 123]. This concept is mainly associated with the use of *tone clusters* in his works, such as *Lux Aeterna* (1966). According to Ligeti, the tone cluster *"is somewhere between sound and noise and consists of several voices stratified and interwoven in semitones, which thereby give up their individuality and become completely dissolved into the resultant overriding complex"* [20, p. 165].

2.2 First movement of the Ten Pieces for Wind Quintet

Ten Pieces for Wind Quintet was composed in 1968, and dedicated to the Wind Quintet of the Royal Stockholm Philharmonic Orchestra. Each movement was conceived as a *micro concerto* with *tutti* odd movements and *solo* even movements, each solo being dedicated to one of the performers [33]. We choose as analytical corpus, for all feature extraction and section division, the version of the piece performed by *London Winds* in the album *Ligeti Edition 7: Chamber Music*, recorded in 1998.

Vitale [33] presents a thorough score-based analysis of this work, and highlights the gradual processes appearing in the piece, based on micropolyphonic strategies to generate the musical texture, where the musical material is articulated with slow modifications in pitch, timbre, density, and rhythm [33, p. 2]. Using as criteria pitch register and dynamics, this author proposes a division of the score of the first movement of the *Ten Pieces for Wind Quintet* (1968) in two main sections (Section 1: measures 1 - 16; Section 2: measures 16 - 22) followed by an appendix (measures 22 - 25). This corresponds respectively to the following time segments in the London Winds recording: 0:00.000 - 1:32.367 (Section 1), 1:32.367 - 1:58.390 (Section 2), 1:58.390 - 2:17.048 (Appendix).

2.3 Audio descriptors

The use of audio descriptors in the context of musical analysis is a multidisciplinary task [36] which admits a multiplicity of approaches depending on the context in which it is applied [21, 23, 36]. In this work, we design the analytical methodology anchored in audio descriptors for two main reasons: 1. audio descriptors provide a perspective (in our case, a perceptual perspective) on the musical sound data, allowing a better understanding of the musical composition [23]; 2. graphical representations of audio

descriptors guide the listening throughout the analysis and facilitates the observation of related perceptual concepts [11]. As detailed in Section 3, we associate the concepts of *timbre of movement* and *permeability* with four audio descriptors: *loudness*, *roughness*, *spectral irregularity* and *spectral entropy*.

Loudness - *Loudness* is a psychoacoustic measure of sound intensity, usually associated with the perception of *dynamics* [3] in musical analysis. The total *loudness* of a time frame (segment of an audio signal) is based on Zwicker's *critical bandwidth* model [37, 7]. The specific *loudness* of each *Bark*[4] band can be computed by a simplification of the original equation [27] as

$$\text{Loudness} = \sum_{z=1}^{N} E(z)^{0.23},$$

where $E(z)$ is the energy in the z-th *bark* band for the time frame considered.

Roughness - According to Vassilakis [31], *roughness* is a perceptual feature related with the sense of very fast amplitude variations in the sound and it is partially conditioned by both the sound stimulus and the properties of the basilar membrane. The *roughness* value of a time frame is based on an approximation, proposed in [30], of the Plomp & Levelt experimental dissonance curve[5] [28]. For complex sounds, the *roughness* value can be computed using a formulation by Vassilakis, which embodies the physical and psychoacoustic mechanisms involved in its perception, as

$$\text{Roughness} = \sum_{i=1}^{N}\sum_{j=i}^{N} \frac{(a_i * a_j)^{0.1}}{2}\left(\frac{2a_j}{a_i + a_j}\right)^{3.11}\left(e^{\frac{0.84|f_j - f_i|}{0.0207 f_i + 18.96}} - e^{\frac{1.38|f_j - f_i|}{0.0207 f_i + 18.96}}\right),$$

where f_i is the i-th partial of the sound and a_i its corresponding amplitude.

Spectral Irregularity - The *spectral irregularity* feature used in this work was proposed by Krimphoff et al. [18] as a measure of the noise content of the spectrum [7, p. 60]. It is usually computed for each time frame in the magnitude spectrum as

$$\text{Irregularity} = \sum_{k=2}^{N-1} \left| a_k - \frac{a_{k-1} + a_k + a_{k+1}}{3} \right|,$$

where a_k is the value in the k-th magnitude coefficient and N is the total number of frequency bins in the spectrum.

A low *irregularity* value denotes a spectrum whose energy is concentrated in few frequency bins, associated with distinguishable components in the sound. In contrast, a high *irregularity* value implies a more regular energy distribution across all frequencies, associated with a more noisy content [7].

Spectral Entropy - *Spectral entropy* is an audio feature used for estimating signal information and complexity in the Time-Frequency Plane [15]. Higher *entropy* values are usually associated with higher spectral activity along all frequencies, and lower values are related to a concentration of spectral energy on few components.

[4] *Bark* is the unit of Zwicker's *critical bandwidth* model [37].

[5] For a full revision on *roughness* curves, see [31].

The *Spectral Entropy* descriptor is derived from Shannon's information theory equation through an analogy between signal energy densities and probability densities [34], as

$$\text{Entropy} = -\sum_{k=1}^{N} P(E_k)\ln(P(E_k)),$$

where $P(E_k)$ denotes the relative frequency of the energy present in the k-th bin.

3 Methodology

Sound mass music, as already pointed, is intrinsically related to the perceptual outcome of the musical events [26, 12, 3]. Therefore, any musical analysis focused only on the score would not provide all relevant information for the understanding of *sound mass* features [3]. It is our aim to devise an appropriate approach to *sound mass* music analysis, based on information extracted from a musical performance through derived audio features. In order to do that, we propose a methodology for *sound mass* music analysis using audio descriptors and psychoacoustic features associated with Ligeti's musical concepts, aligned with score-based information.

The concept of *timbre of movement* is associated with the dynamic perception of the global behavior of musical texture and its *microtime* manifestation [29, 1]. The *loudness* descriptor is used as a measure for *global perceptual dynamics* [3] and the *roughness* descriptor was used to describe the *microtime behavior* of *sound masses* [1, 31].

According to the principles of *minimum masking* and *limited density* [17], the higher the level of spectral information in the auditory nerves, the lower our ability to perceive musical pitches and intervals. Therefore, the concept of *permeability* is represented by the *textural information level* of the *sound mass*, associated with *spectral entropy* [4], and the *noise content*, associated with *spectral irregularity* [7].

Based on the above concepts and their interrelationships, the analysis was conducted in 3 steps: 1. Manual segmentation of the audio signal according to the score, as described in Section 2.2; 2. Computation of the selected time-varying audio descriptors (Section 2.3), their corresponding graphical representations, mean and standard deviation values, as well as scatter plots to illustrate their correlations; 3. Musicological (human conducted) analysis of the piece to establish the relationships between musical content of the different sections and the obtained descriptors values.

Feature extraction was done using Python[6] and the Jupyter[7] environment. *Loudness*, *irregularity* and *entropy* were computed from the magnitude spectrogram obtained using Librosa [24], with a window size of 4096 samples and hop length of 1024 samples. The *roughness* descriptor was obtained from the reassigned spectrogram [16] (with the same parameters described above), as it depends on precise frequency and amplitude values. All the code used to extract and plot the audio features is available at a Gitlab repository[8].

[6] `https://www.python.org/`
[7] `https://jupyter.org/`
[8] `https://gitlab.com/Feulo/ligetis-wind-quintet-analysis`

4 Results and Discussion

Figure 1 corresponds to the graphical representations of the descriptors obtained from the audio analysis as functions of time. Each different color represents one of the three sections of the score: blue is the first section, orange the second section and green the appendix. Mean and standard deviation values for each descriptor and section are presented in Table 1. The analysis was conducted in 2 stages: first we explore the characteristics associated with *timbre of movement*, followed by the behavior associated with *permeability*.

Timbre of movement - According to the methodology proposed (Section 3), the psychoacoustic descriptors of *loudness* and *roughness* are associated with the concept of *timbre of movement*. Each section of the piece displays a different behavior in terms of this concept.

The first section displays a somewhat regular fluctuation of *loudness* values (blue line in the upper left corner of Figure 1). Within the same section, *roughness* (blue line in the upper right corner of Figure 1) displays low values with low variation. The corresponding statistics can be seen in Table 1.

In evident contrast with the first section, the second section of the piece displays the highest values of *loudness*. We highlight that, although the standard deviation values for these two sections are not very different, by inspection of the *loudness* curve, we can see that the first section has an oscillatory behavior while the second section displays an ascending pattern. Also, in the second section we observe the highest values of *roughness* with a complex oscillatory pattern, with spikes that go upwards towards the end of this section. Finally, the appendix presents low values and low variation for both *loudness* and *roughness*.

Section	*Loudness*	*Roughness*	*Irregularity*	*Entropy*
1	41.87 ± 14.49	9.96 ± 5.86	63.54 ± 40.72	0.42 ± 0.13
2	66.36 ± 12.52	103.12 ± 62.44	171.63 ± 58.37	0.74 ± 0.15
3	13.36 ± 5.29	0.55 ± 0.56	9.98 ± 6.58	0.10 ± 0.03

Table 1. Mean and standard deviation values for *loudness* and *roughness* on each section.

Permeability - Ligeti's concept of *permeability* is linked to the audio descriptors of *spectral irregularity* and *spectral entropy*. By observing the two curves at the lower half of Figure 1, we can also observe a distinct profile within each one of the sections of the piece.

In the first section, a regular fluctuation of the values is observed in both descriptors, but *entropy* displays a lower range of variation relative to the mean ($\sigma/\mu = 0.64$ for *irregularity* and $\sigma/\mu = 0.31$ for *entropy*, according to the values in Table 1). In the second section, we can observe an ascending pattern in both features, similarly to what was observed for *loudness* and *roughness*, with increasing spikes in the *spectral entropy* profile. The same observation can be made here for the relative variation of both features, with $\sigma/\mu = 0.34$ for *irregularity* and $\sigma/\mu = 0.20$ for *entropy*. Finally, the appendix displays once again the lowest values in both descriptors, as observed with *loudness* and *roughness*, with smaller relative *entropy* variation ($\sigma/\mu = 0.30$) with respect to irregularity ($\sigma/\mu = 0.66$).

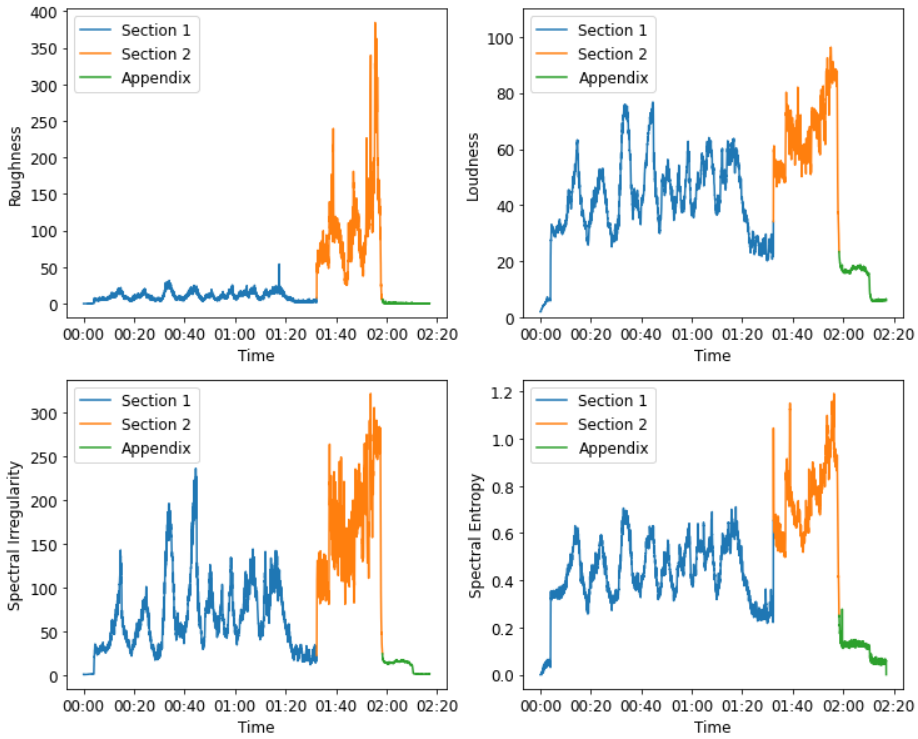


Fig. 1. *Roughness* (upper left), *Loudness* (upper right), *Spectral Irregularity* (lower left) and *Spectral Entropy* (lower right) for the 3 sections of the piece.

By observing all descriptors taken together, we see that the three sections of the piece have very different behaviors from the perspective of their perceptual features. The psychoacoustic difference between the sections is illustrated in the first plot of Figure 2, which represents the relationship between *loudness* and *roughness*, and in the correspondence between *spectral irregularity* and *spectral entropy*, shown in the second plot of Figure 2. In both graphs, the spatial placement of the three section clusters, as well as their geometrical arrangement, make the exploration of *timbre of movement* and *permeability* relatively explicit, allowing us to observe a link between the formal division of the composition and the different perceptual feature aspects of the sound material.

Section 1 has a focus on the global behavior of the musical texture in terms of *timbre of movement*, with a high variation of *permeability*. Section 2 emphasizes *timbre of movement* with a focus on the *microtime* behavior, while at the same time reaching the highest levels of *permeability*. The appendix displays a low level of activity in terms of both *timbre of movement* and *permeability*. It is interesting to observe that, with respect to the first section, we see the blue cluster lying horizontally on the scatter plot, where the large variations in *loudness* emphasize the global dynamic perception. In contrast, the second section (orange) corresponds to a highly scattered cluster in both *roughness* and *loudness* axes, but concentrating on high values of *loudness*, thus bringing the *mi-*

crotime behavior (variation of *roughness*) to the forefront. In terms of *permeability*, we

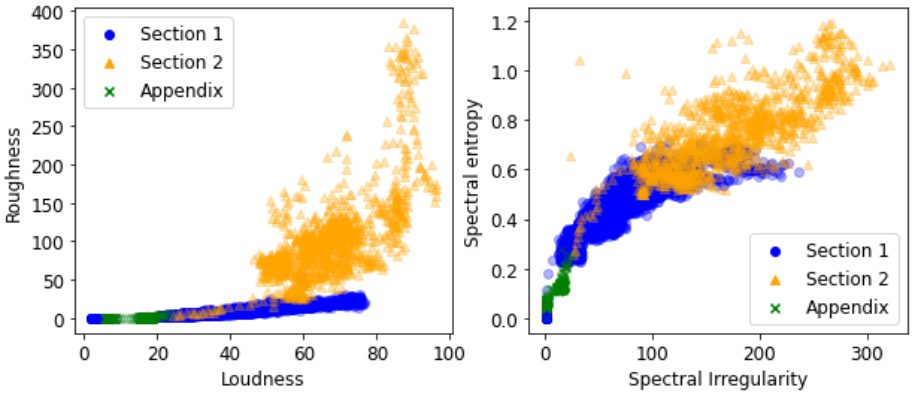


Fig. 2. Scatter plot with the *loudness* (x-axis) and *roughness* (y-axis) values (left). Scatter plot with the *spectral irregularity* (x-axis) and *spectral entropy* (y-axis) values (right)

can observe in Figure 2 that the *spectral irregularity* and *spectral entropy* fluctuations in all sections are highly correlated, producing a log-like, quasi-diagonal shape in the scatter plot, which correspond to constant changes of pitch perception in the musical texture. This might be associated with the harmonic technique of *blurring* [8, p. 122], used by Ligeti to manipulate the musical texture. Especially in the second section (orange), the higher values of *loudness* and the large variation of the spectral descriptors obliterate the perception of individual events, turning our attention to the mass behavior of the composition. Finally, it is interesting to notice that the low level of all descriptors in the third part (green) could be the reason why Vitale [33] described this section as an appendix of the piece.

5 Conclusion

In this paper we presented a methodology for the musical analysis of *sound mass* compositions, based on audio descriptors associated with Ligeti's concepts of *timbre of movement* and *permeability*. In terms of the musicological interest in audio analysis techniques focused on *sound mass* composition, the proposed method reinforces the idea that the perceptual features associated with the performance of a musical work bring important elements that help understanding the formal development of a work, without reducing the importance of symbolic analyses based on the musical score. By analyzing the psychoacoustic features of each section of this particular piece, we may argue that the most important perceptual characteristics of the work do not depend heavily on specific choice of pitches, rhythms or harmonies, but are highly anchored on the perceptual qualities of the *sound masses*. Also, audio descriptors could expand the *gradual process* approach [32], enriching the symbolic analysis with performative characteristics of the piece.

Future work may focus on investigating other timbre-related psychoacoustic descriptors in the context of the proposed analysis, to verify whether they contribute to

a better understanding of the perception of similarity of *sound masses* [22]. The study of other audio descriptors in the context of *sound mass music* could also foment applications in the field of creative processes, particularly in computer-aided composition and musical modeling. Exploration of perceptual features of a musical work through comparative analysis of different recordings of the same piece is also an interesting avenue for future work. It would be useful to investigate how the interpretative choices in different performances could reveal the invariant properties of a musical work [35], as expressed in the score. Finally, musical analysis with audio descriptors might help in empirical studies with musical excerpts [12], offering exploratory ways to represent perception attributes of non-expert listeners.

References

1. Antunes, M., Feulo, G., Manzolli, J.: A perceptual approach to Ligeti's Continuum with a Roughness descriptor. In: Livro de Resumos do II EINEM Encontro Internacional de Investigação de Estudantes em Música e Musicologia. pp. 18–19. Évora (2020)
2. Antunes, M., Manzolli, J.: A psychoacoustical approach to Ligeti's concept of permeability. In: Livro de Resumos do II EINEM Encontro Internacional de Investigação de Estudantes em Música e Musicologia. pp. 20–21. Évora (2020)
3. Antunes, M., Rossetti, D., Manzolli, J.: Emerging structures within micro-time of Ligeti's Continuum (pre-print). In: Proceedings of the 2021 International Computer Music Conference. Santiago, Chile (2021)
4. Baraniuk, R.G., Flandrin, P., Janssen, A.J., Michel, O.J.: Measuring time-frequency information content using the rényi entropies. IEEE Transactions on Information theory 47(4), 1391–1409 (2001)
5. Borio, G., Danuser, H.: Die Internationalen Ferienkurse für Neue Musik Darmstadt 1946-1966. Concert program, lectures, masterclasses, tutors. In: Geschichte und Dokumentation in vier Bänden., vol. 3. Rombach (1997)
6. Bregman, A.S.: Auditory scene analysis: The perceptual organization of sound. MIT press (1994)
7. Bullock, J.: Implementing audio feature extraction in live electronic music. Ph.D. thesis, Birmingham City University (2008)
8. Cambouropoulos, E., Tsougras, C.: Auditory Streams in Ligeti's Continuum: A Theoretical and Perceptual Approach. Journal of interdisciplinary music studies 3(1-2), 119–137 (2009)
9. Clendinning, J.P.: Contrapuntal techniques in the music of Gyorgy Ligeti. Ph.D. thesis, Yale University (1990)
10. Clendinning, J.P.: The Pattern-Meccanico Compositions of György Ligeti. Perspectives of New Music 31(1), 192 (1993)
11. Couprie, P.: Graphical representation: an analytical and publication tool for electroacoustic music. Organised Sound 9(1), 109–113 (2004)
12. Douglas, C., Noble, J., McAdams, S.: Auditory Scene Analysis and the Perception of Sound Mass in Ligetis Continuum. Music Perception 33, 287–305 (2016)
13. Fastl, H., Zwicker, E.: Psychoacoustics: facts and models. No. 22 in Springer series in information sciences, Springer, Berlin ; New York, 3rd. ed edn. (2007)
14. Ferraz, S.: Análise e Percepção Textural: Peça VII, de 10 peças para Gyorgy Ligeti. Cadernos de Estudos pp. 68–79 (1990)
15. Figueiredo, N.S.: Efficient adaptive multiresolution representation of music signals. Master dissetation, University of São Paulo (2020)

16. Fulop, S.A., Fitz, K.: Algorithms for computing the time-corrected instantaneous frequency (reassigned) spectrogram, with applications. The Journal of the Acoustical Society of America 119(1), 360–371 (2006)
17. Huron, D.: Tone and voice: A derivation of the rules of voice-leading from perceptual principles. Music Perception: An Interdisciplinary Journal 19(1), 1–64 (2001)
18. Krimphoff, J., McAdams, S., Winsberg, S.: Caractérisation du timbre des sons complexes.II. Analyses acoustiques et quantification psychophysique. Le Journal de Physique IV 04(C5), C5–625–C5–628 (May 1994)
19. Ligeti, G.: Neuf essais sur la musique. Éditions Contrechamps, Genève – Suisse (2010)
20. Ligeti, G., Bernard, J.W., Ligeti, G.: States, Events, Transformations. Perspectives of New Music 31(1), 164 (1993)
21. Malloch, S.N.: Timbre and Technology. PhD Thesis, The University of Edinburgh, Edinburgh (1997)
22. McAdams, S.: Timbre as a structuring force in music. In: Timbre: Acoustics, perception, and cognition, pp. 211–243. Springer (2019)
23. McAdams, S., Depalle, P., Clarke, E.: Analyzing musical sound. In: Empirical musicology: Aims, methods, prospects, pp. 157–196. Oxford University Press, Oxford (2004)
24. McFee, B., Raffel, C., Liang, D., Ellis, D., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and Music Signal Analysis in Python. In: 14th Python in Science Conference. pp. 18–24. Austin, Texas (2015)
25. Misra, H., Ikbal, S., Bourlard, H., Hermansky, H.: Spectral entropy based feature for robust asr. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 1, pp. I–193 (2004)
26. Noble, J., McAdams, S.: Sound mass, auditory perception, and 'post-tone' music. Journal of New Music Research 49(3), 231–251 (May 2020)
27. Peeters, G.: A large set of audio features for sound description (similarity and classification) in the CUIDADO project. CUIDADO IST Project Report 54(0), 1–25 (2004)
28. Plomp, R., Levelt, W.J.M.: Tonal consonance and critical bandwidth. The journal of the Acoustical Society of America 38(4), 548–560 (1965), publisher: Acoustical Society of America
29. Roads, C.: Microsound. MIT Press, Cambridge, Mass (2001)
30. Sethares, W.A.: Tuning, timbre, spectrum, scale. Springer Science & Business Media, Berlin (1998)
31. Vassilakis, P.N.: Perceptual and physical properties of amplitude fluctuation and their musical significance. PhD Thesis, University of California, Los Angeles, Califórnia (2001)
32. Vitale, C.: A gradação nas peças 5 e 6 das Dez peças para quinteto de sopros de György Ligeti. In: Anais do I Encontro Internacional de Teoria e Análise Musical. pp. 1 – 8 (2009)
33. Vitale, C.H.: Dez peças para quinteto de sopros de György Ligeti: a gradação como uma ferramenta para a construção do discurso musical. PhD Thesis, Universidade de São Paulo (2008)
34. Williams, W.J., Brown, M.L., Hero III, A.O.: Uncertainty, information, and time-frequency distributions. In: Advanced Signal Processing Algorithms, Architectures, and Implementations II. vol. 1566, pp. 144–156. International Society for Optics and Photonics (1991)
35. Ystad, S., Aramaki, M., Kronland-Martinet, R.: Timbre from sound synthesis and high-level control perspectives. In: Timbre: Acoustics, Perception, and Cognition, pp. 361–389. Springer (2019)
36. Zattra, L.: Analysis and analyses of electroacoustic music. In: Proceedings of the Sound and Music Computing 2005. p. 10. Salermo, Italy (2005)
37. Zwicker, E., Flottorp, G., Stevens, S.S.: Critical Band Width in Loudness Summation. The Journal of the Acoustical Society of America 29(5), 548–557 (May 1957)

Unsupervised method for Implementing Implication-Realization Model Analyzer on Computer

Kaede Noto[1], Yoshinari Takegawa[1], and Keiji Hirata[1⋆]

Future University Hakodate
g3120002@fun.ac.jp

Abstract. We propose an implementation method for an implication-realization (I-R) model analyzer that is based on note duration, beat structure, and pitch transition. The proposed method involves two procedures, i.e., symbol-start note estimation based on note duration, beat structure, and pitch transition and symbol assignment by introducing a method for recurrently determining small and large intervals in an I-R model analyzer. With the Narmour's manual I-R analysis, our methof had an F measure of 0.86 symbol-start note estimation.

Keywords: Implication Realization Model, Music theory, Music cognition

1 Introduction

We propose an implementation method for an implication-realization (I-R) model [2],[3] analyzer that is based on note duration, beat structure, and pitch transition. I-R analysis classifies the relationship between adjacent notes in accordance with how implications are satisfied or denied. The method for determining these relationships is based on Gestalt theory. In Gestalt theory, the perceptual elements are grouped and recognized. Narmour claims that there is a similar principle in the perception of melody. The smallest unit of a group by I-R analysis is three notes, which are assigned symbols in accordance with their characteristics. For example, the symbol P (process) is assigned to a sequence of notes that are implicated to be heard at the same interval in the same direction. Even when the same implication occurs, the symbol IP (intervallic process) is assigned when only the implication of interval is satisfied, and the symbol VP (registral process) is assigned when only the implication of direction is satisfied. Thus, in I-R analysis, symbols are assigned in terms of whether the melodic expectation is satisfied or denied concerning interval or direction.

I-R analysis involves two procedures. First, to obtain I-R analysis results, it is necessary to estimate the I-R symbol-start note. This is an operation to discover the cognitive boundary in the melody. According to Narmour, the clue to estimating the symbol-start note is the "closure". A closure is a note at which no implication arises from the sequence of notes occurring or where the implication is weakened. In other words, a closure is an event that triggers a grouping boundary. Specifically, it refers to a change in pitch interval, direction, or note value or the occurrence of a strong beat, etc. I-R analysis is conducted for triplets starting from those boundaries.

⋆ Please place acknowledgement here.

Second, the I-R symbols are assigned to a sequence of notes that is longer than three notes, starting with the note estimated in the first procedure. Symbols are assigned mainly on the basis of two principles, principle of intervallic difference (PID) and principle of registral direction (PRD). The PRD states that small (five seminotes or less) intervals imply an interval in the same direction, and large (seven seminotes or more) intervals imply an interval in the registral direction. The PID states that small intervals imply a similarly-sized (plus or minus two seminotes) interval, and large intervals imply a small interval. On the basis of these principles, three particular notes are assigned one of eight symbols. By completing these two procedures, we can obtain the results of I-R analysis.

The purpose of this study was to develop an implementation method for an I-R model analyzer and quantitatively evaluate the results of the analysis. Although several methods have been proposed to implement an I-R model analyzer, there have not been studies that have quantitatively evaluated the accuracy of these methods. Grachten et al. proposed an implementation method for an I-R model analyzer using decision trees [4], and Yazawa et al. proposed one using extended I-R symbols [8], but to evaluate these methods, they used the performance of melodic similarity with the analysis results as features.

Because the performance of these methods are not known. Therefore, the usefulness of the I-R analysis results in the Music information retrieval (MIR) field is not clear. In this study, we evaluated the performance of our proposed method by comparing it with the Narmour's manual analysis.

2 Methodology

The flow of our method is as follows (Figure 1). First, we estimate the symbol-start note. This consists of two steps: closure estimation (Section 2.1, 2.2) and determining the order of symbol assignment (Section 2.3, 2.4, 2.5). Second, after the symbol-start note is estimated, we assign it a symbol (Section 2.6, 2.7).

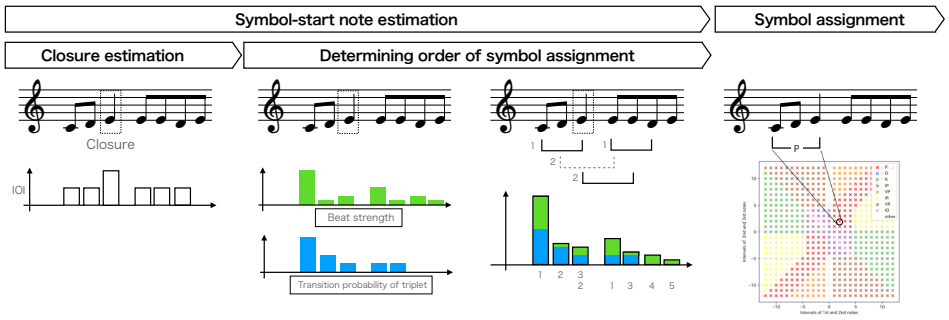


Fig. 1. Flow of proposed method

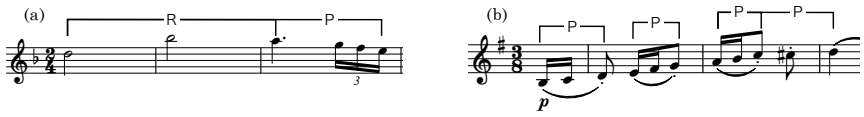


Fig. 2. I-R analysis by Narmour. (a) W. A. Mozart: Piano Concerto No. 19 in F Major, Kv.459, 3rd mov., (b) L. v. Beethoven: Piano Sonata op.14-2, 3rd mov..

2.1 Closure

We first explain a closure to estimate the starting note of the symbol. We give examples of the I-R analysis by Narmour (Figure 2) to explain the relationship between a closure and symbol-start note. In the example in Figure 2 (a), the note considered to be the closure is the first note of the third measure, i.e., "la," which has a strong beat and changing note value. In this case, the closure "la" is the end of symbol R, which starts in the first measure, and the beginning of symbol P, which starts in the third measure.

In the example in Figure 2(b), the first and fourth notes of the first measure, third note of the second measure, and first note of the third measure are considered closures. In this case, the first, fourth, and third notes of the first measure are the end notes of symbol P, and the third note of the second measure is the start and end notes of the symbol. Thus, we define "closure" as the union set $A \cup B$ when the symbol-start note is represented by set A and the symbol-end note by set B.

2.2 Closure Estimation Based on Inter-onset Interval

Our proposed method uses a closure-estimation method that focuses on the change in the inter-onset interval (IOI). Researchers have attempted to estimate a closure by focusing on note duration. For example, with current closure-estimation methods, the closure is considered to be the point where the note duration increases [6] or rests occur [8]. The problem with these methods is that they estimate many closures for melodies that have alternating notes and rests. However, we integrate the above methods by focusing on the change in the IOI. Because the IOI is the difference between the times at which each note occurs, the IOI of any two notes will not change even if the note value changes or rests are inserted, unless the timing of onset is changed. Because time is handled differently for note value and IOI, it is necessary to develop a method for detecting changes in note duration when targeting the IOI. Current methods are based on the note value (quarter notes, eighth notes) in a score. It is reasonable that an increase in note value is defined as a factor of two or more compared with the previous notes. Therefore, we consider a note at which the IOI increases by a factor of two or more compared with the previous IOI as the closure.

Closure estimation can be used to limit the targets for symbol assignment. As above, because we define a closure to be the union set of the start and end notes of a symbol, we do not assign symbols across closures. However, if the symbols before and after the closure are identical, sometimes they may be regarded as symbols across the closure. The details are given later.

2.3 Transition of Pitch

We introduce transition of pitch on the basis of the hypothesis that the first note of a pitch-transition pattern that frequently occurs in a melody is likely to be the symbol-start note. Because the length of the I-R symbol is three, we calculate the probability of occurrence of a tri-gram in a melody. In addition, the first two notes in the I-R symbol are those that generate an implication. The third note will satisfy or deny the implication. Thus, we use the probability of the $(i+2)$ th note occurring after the $(i+1)$th and i th notes, $P(i+2|i, i+1)$, as a feature for estimating the symbol-start note.

The distribution of $P(i+2|i, i+1)$ changes depending on how the random variable is determined. For example, when determining the transition probability of a melody, the pitch is generally used as a random variable. However, when the pitch is used as a random variable, the probability distribution after learning is likely to be sparse. To avoid this, we consider two random variables, pitch interval and qualitative pitch interval. Because pitch intervals are divided into two values, i.e., S (small) and L (large), in I-R analysis, we define qualitative pitch as a binary expression of n or less seminotes and more than n seminotes.

2.4 Beat Structure

We use the beat structure for symbol assignment. The beat structure is known to affect group formation when listening to a melody. Fraisse reported that when presented with a sequence of sounds that occur in the same time span, people divide these sounds into two or three repetitive groups [7]. We hypothesize that the I-R symbols are also a type of group, and that symbols are assigned on the basis of the beat. For beat strength, we use the value obtained from Music21Object.beatStrength implemented in the Python library music21 [1] as a feature value. In the beatStrength object, beat strength is expressed as a relative value, such as 1.0 for the first beat of a measure, 0.5 for downbeats, and 0.25 for upbeats.

2.5 Feature Integration

We estimate symbol-start notes from closure, pitch-transition pattern, and beat structure. The search range for estimating the symbol-start note is the interval from one closure to the next. We integrate pitch-transition pattern and beat structure within this interval. Integration refers to standardizing each value then calculating the sum. Because the sum of values indicates how likely it is to be a symbol-start note, we assign the symbols in order, starting with the highest value.

3 Symbol Assignment

3.1 Previous Symbol-assignment Method and Actual Data

There are ambiguities with current method of I-R symbol assignment. Basically, we can make rules from the PID and PRD proposed by Narmour on how to assign symbols to the three notes. However, we also need to determine a threshold for determining the S

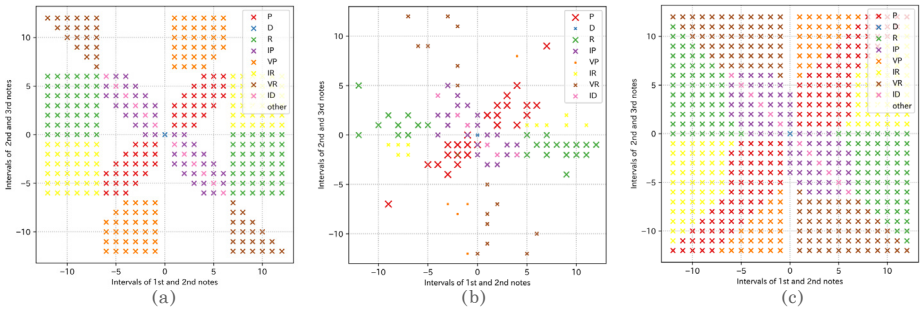


Fig. 3. Distribution of I-R symbols assigned to triplet. Horizontal axis is number of seminotes in first and second notes, and vertical axis is number of seminotes in second and third notes.

or L pitch interval as a hyperparameter. Figure 3(a) shows the distribution of symbols when S is six or less seminotes and L is more than six seminotes, indicating that different symbols are being assigned after the threshold. Nothing is written to the coordinates corresponding to triplets that do not assign the I-R symbols. Figure 3(b) shows the distribution of the I-R symbols observed from Narmour's manual analysis. These figures show the correspondence between the symbols observed from Narmour's analysis and the pitches of the triplets, with the size of each point proportional to the number of times it was observed. We did not observe any example of "other" symbol assigned to a triplet. The intricacy of each symbol's region concerning the axial direction suggests that the threshold for determining the I-R symbol to be assigned is not fixed.

We introduce a symbol-assignment method that recursively changes the threshold. If the threshold is defined as n, then S can be regarded as a pitch within n semitones, and L as a pitch greater than n semitones. The initial threshold is n = 6, following Narmour's rule-based method. After the symbol assignment with n = 6 is completed, we gradually increase the threshold from n = 1. This operation yields a distribution of symbols, as shown in Figure 3(c). We can see that it includes the symbols in Figure 3(a) and many of those in Figure 3(b).

3.2 Detailed Rules for I-R Symbol Assignment

To conduct I-R analysis for an actual melody, we need to determine the number of symbols to be assigned to each note. For example, if we allow three symbols to be assigned to any note, the operation is the same as the I-R analysis for a tri-gram. We conducted I-R analysis with two and three maximum symbol assignments and compared the results.

The object of I-R analysis is a triplet, but four or more notes may be assigned symbols P or D (Duplicate). These symbols have the characteristic of repeating similar pitches in the same direction. Thus, the repetition of symbol P or D is thought to amplify the implication. Therefore, if symbols P and D are superimposed, they are merged and considered one symbol.

However, this is not the case if the implied attenuation occurs within symbols P and D, which consist of four or more notes. As already indicated, even when symbol P is consecutively, they may not be integrated (Figure 2). This is thought to be due to the fact that closures occur between symbols. However, it is difficult to investigate all the possibilities of generating closures. Thus, we introduce a symbol-integration rule that focuses only on the beat structure, which can be understood intuitively.

4 Experiment

We conducted an evaluation experiment to investigate the accuracy of the proposed implementation method in estimating the symbol-start tone and the factors that contributed to the results. There were five evaluation items.

Table 1. Evaluation items

Evaluation items	
1. Features	1-1. Closure estimated from IOI 1-2. Transition probability of triplet 1-3. Beat strength
2. Random variable with transition probability of triplet	2-1. Pitch 2-2. Interval 2-3. Qualitative pitch interval
3. Maximum number of symbols assigned	3-1. Two 3-2. Three
4. Division of symbols (Beat Strength)	4-1. No division 4-2. First beat of measure (1.0) 4-3. Downbeat (0.5) 4-4. Downbeat and upbeat (0.25)
5. Threshold of symbol assignment	5-1. Narmour's method (N = 6) 5-2. Proposed Method

4.1 Evaluation values and dataset

We used the results of the manual analysis by Narmour as the correct data. Because the rules of I-R analysis are often ambiguous and the results are subjective, there is no large data set of I-R analysis results. Thus, we used 61 examples taken from Narmour's analysis examples [3] as the correct data. The melodies used as the correct data were selected on the basis of the following two criteria. The first criterion is the number of notes contained in the melody to be analyzed. If the number of notes is four or less, the results of the I-R analysis can be uniquely determined. Thus, we did not take into account melodies not considered as correct data. The second criterion is whether the three tones to be analyzed are adjacent to each other. In Narmour's analysis, there is

an example in which similar sound sequences are considered as one cohesive unit, and the beginning of the unit is extracted and subjected to I-R analysis. In these cases, we did not consider them as correct data because it is necessary to select three tones for I-R analysis, which is beyond the scope of this study. The correct data were all created manually as MusicXML with the following information: pitch, duration, onset time, I-R symbol, and symbol-start note.

The input to the system is pitch, duration, beat strength, and the output is a binary value indicating whether the note is a symbol-start note. Thus, we used recall, precision, and F-measure to evaluate the method.

4.2 Training Data

To calculate Feature 1-2. (Transition probability of triplet), we used 300 melodies from GTTM DataBase [5] as training data. As mentioned above, we did not label the training data because what we want is the conditional probability of three adjacent notes in the melody. Because the conditional probability will be zero if the pitch-transition pattern included in the target melody does not exist in the training data, we included the target melody in the training data.

4.3 Evaluation Results

Figure 4 presents the results for symbol-start note estimation when different features are used for estimation. Cases 1 to 8 on the horizontal axis of each bar graph correspond to the combinations of features in Table 2. The bars located on the left side are the evaluation scores when more features were used. Case 1 is the result of estimation with three features, and Case 8 is that without any features. The highest score was obtained when all the features were used.

We found that the score tended to increase with the number of features used. However, there was no difference in the F-measures between Case 4, which used two features, and Case 5, which used one feature. Because the difference between Cases 4 and 5 is the presence or absence of Feature 1-2, Feature 1-2 is considered to have an effect on the score. However, there was a difference of 0.1 in the F-measure of Cases 2 and 6, which also differed only in the presence and absence of Feature 1-2. This result indicates that it is not only the features used in the estimation but also the combination of features.

Table 2. Feature selection

Feature	Evaluation items							
	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6	Case 7	Case 8
1-1. Closure estimated form IOI	✓		✓	✓	✓			
1-2. Transition probability	✓	✓		✓			✓	
1-3. Beat strength	✓	✓	✓			✓		

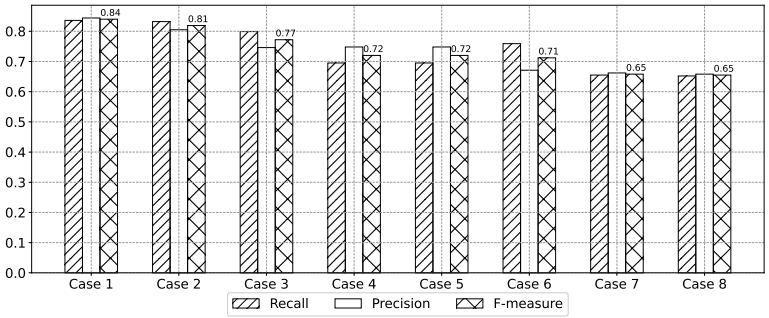


Fig. 4. Experimental results for Cases 1 − 8 regarding 2-3. Qualitative pitch interval, 3-2. Three, 4-2. First beat of measure, and 5-1. Narmour's method (N = 6)

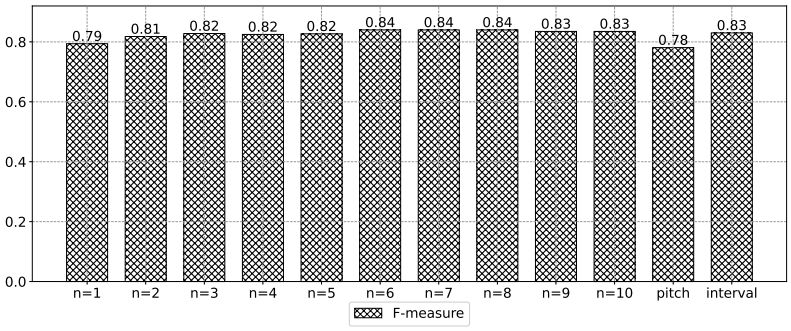


Fig. 5. Experimental results for evaluation item 2 regarding Case 1, 3-2. Three, 4-2. First beat of measure and 5-1. Narmour's method (N = 6))

Figure 5 presents the results for symbol-start note estimation when conditional probabilities are calculated using different random variables. The value of n in the graph represents the number of semitones used to determine the qualitative pitch interval. For example, if n = 3, all intervals appearing in the melody are represented as two values, one for intervals of three semitones or less, and one for intervals of four semitones or more. The highest evaluation values were obtained when n = 6, 7, and 8.

Figure 6 presents the results for symbol-start note estimation when comparing the maximum number of symbols assigned. When the maximum number of symbols assigned is three (3-1.), the analysis results are equivalent to the I-R analysis results for a tri-gram with the symbols that straddle the closure removed. We can see in Figure 6 that when the maximum number of symbols is three (3-1.), recall is higher than when the maximum number of symbols is two (3-2.). This is because when the maximum number of symbols is three, our method estimates more symbol-start notes. However, precision decreased. Therefore, Figure 6 indicates that if we want to achieve a high F-measure, it is better to use the maximum number of symbols of two.

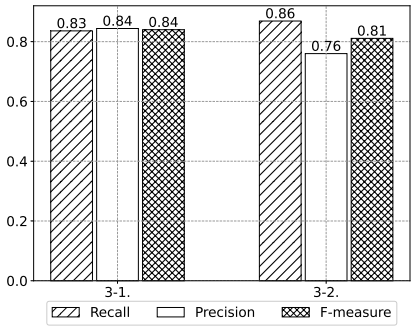


Fig. 6. Experimental results for evaluation item 3 regarding Case 1, 2-3 Qualitative pitch interval, 4-2. First beat of measure and 5-1. Narmour's method (N = 6)

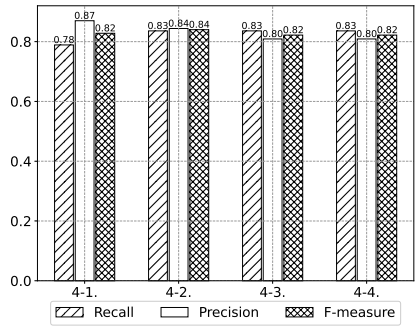


Fig. 7. Experimental results for evaluation item 4 regarding Case 1, 2-3. Qualitative pitch interval, 3-2. Three, and 5-1. Narmour's method (N = 6)

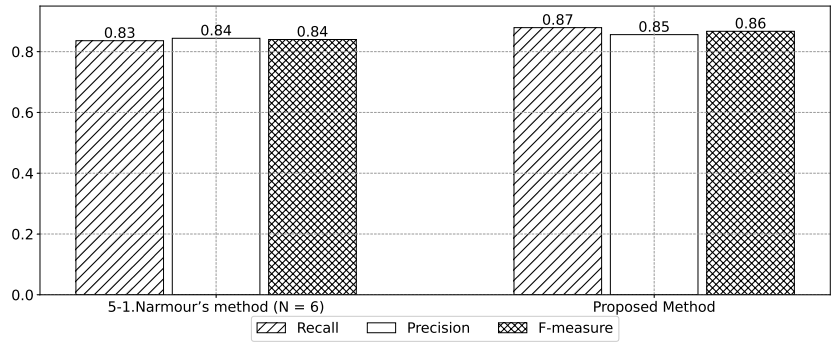


Fig. 8. Experimental results for evaluation item 5. regarding Case 1, 2-3. Qualitative pitch interval, 3-2. Three, and 4-2. First beat of measure

Figure 7 presents the results for symbol-start note estimation when symbols P and D, which consist of four or more notes, are divided in accordance with the beat strength. Thus, the number of notes considered to be symbol-start notes increased. The condition with the highest precision was when no splitting was carried out. However, recall was lowest among the four conditions, which indicates that the coverage in finding the symbol-start note is low. The highest F-measure was obtained when beat strength was 4-1. (beat strength = 1.0), which is when the symbols are split at the beginning beat of the measure. Also, when splitting symbols on smaller beats (downbeat or downbeat and upbeat), precision decreased. Hence, if we want to increase the accuracy of symbol-start note estimation, only splitting symbols at the beginning of the measure is sufficient.

However, the small difference in the evaluation values for 4-1., 4-2., and 4-3. (beat strength = 1.0, 0.5, and 0.25) may be due to a bias in the appearance of symbols P and D. In this experiment, the best score was obtained by estimating the symbol-start note

with a beat strength of 1.0, but we do not know whether similar results can be achieved when we conduct I-R analysis on melodies with fast passages.

Figure 8 presents a comparison of the results of symbol-start note estimation with different symbol-assignment methods. The method of assigning these symbols conforms to the symbol distribution shown in Figure 3. Narmour's method (N = 6) corresponds to Figure 3(a), and the proposed method corresponds to Figure 3(c). The proposed method had a better score than Narmour's method.

The difference between the two methods is in the handling of symbols that were considered as "other" with Narmour's method. With this method, no symbol is assigned to the triplet corresponding to the "other", but with the proposed method, a symbol is assigned to the triplet. Thus, more notes will be inferred as symbol-start notes with the proposed method.

5 DISCUSSION AND CONCLUSION

We proposed an implementation method for an I-R analyzer with symbol-start note estimation that is based on note duration, beat structure, and uses pitch transition and a symbol-assignment method for changing the threshold recursively. With the examples of Narmour's I-R analysis, our method had an F measure of 0.86. We also conducted a comparative verification for each feature.

We evaluated the accuracy of our I-R analyzer, but its usefulness in MIR is not clear. Features based on human cognition have been used to get boundary, such as lyrics and syllables [9]. By making comparisons with such studies, we hope to be able to compare the usefulness of I-R model from a cognitive perspective. Future work includes feature design for treating I-R symbols as features and comparison with previous studies.

Acknowledgments This work has been supported by JSPS Kakenhi 16H01744.

References

1. Cuthbert, M. S., Ariza, C.: music21: A toolkit for computer-aided musicology and symbolic music data (2010).
2. Narmour, E.: The Analysis and Cognition of Basic Melodic Structures. The University of Chicago Press (1990)
3. Narmour, E.: The Analysis and Cognition of Melodic Complexity. The University of Chicago Press (1992)
4. Grachten, M., Arcos, J. L., and López de Mántaras: Melody retrieval using the implication/realization model. MIREX (2005).
5. Hamanaka, M., Hirata, K., and Tojo, S.: "Musical structural analysis database based on GTTM." (2014)
6. Hatano Y.: Music and Cognition, University of Tokyo Press, pp. 1 − 40 (1989)
7. Paul Fraisse: Rhythm and tempo. The psychology of music, 1, pp.149-180. (1982)
8. Yazawa, S., Hamanaka, M., Utsuro, T.: "Subjective Melodic Similarity based on Extended Implication-Realization Model." IJAE 15.3, pp. 249-257 (2016)
9. Bas, C., Zuidema, W., and Burgoyne, A.: "Mode Classification and Natural Units in Plainchant." Proceedings of the 21th International Conference on Music Information Retrieval (ISMIR 2020). Montréal, Canada. (2020)

Time-span Tree Leveled by Duration of Time-span

Masatoshi Hamanaka[1], Keiji Hirata[2] and Satoshi Tojo[3]

[1] RIKEN
[2] Future University Hakodate
[2] JAIST
masatoshi.hamanaka@riken.jp

Abstract. This paper describes a time-span tree leveled by the length of the time span. Using the time-span tree of the Generative Theory of Tonal Music, it is possible to reduce notes in a melody, but it is difficult to automate because the priority order of the branches to be reduced is not defined. A similar problem arises in the automation of time-span analysis and melodic morphing. Therefore, we propose a method for defining the priority order in total order in accordance with the length of the time span of each branch in a time-span tree. In the experiment, we confirmed that melodic morphing and deep learning of time-span tree analysis can be carried out automatically using the proposed method.

Keywords: Generative theory of tonal music (GTTM), time-span tree, time-span reduction, melodic morphing, Transformer.

1 Introduction

Our goal is to automate the system using a time-span tree of the Generative Theory of Tonal Music (GTTM) [1]. GTTM consists of grouping structure analysis, metrical structure analysis, time-span tree analysis, and prolongational tree analysis. a time-span tree is a binary tree with a hierarchical structure that describes the relative structural importance of notes that differentiate the essential parts of the melody from the ornamentation.

The time-span tree in Fig. 1 is the result of analyzing a melody (a) on the basis of GTTM. Reduced melodies can be extracted by cutting this time-span tree with a horizontal line and omitting the notes connected below the line (Fig. 1 (b)–(f)). Melody reduction with GTTM is the absorption of notes by structurally important notes.

The problem with previous systems using time-span trees is that the priority order of branches of a time-span tree is not defined. The GTTM-based melodic-morphing algorithm we previously proposed was difficult to automate because it included a time-span reduction process [2, 3]. We have been developing a GTTM analyzer using deep learning and have been able to automate grouping structure analysis and metrical structure analysis using deep leaning [4, 5]. How-ever, deep learning of time-span tree analysis is difficult to automate due to the ambiguity of the reduction process.

Therefore, we propose a method for defining the priority order in total order in accordance with the length of the time span of each branch in the time-span tree, ena-

bling melodic morphing and time-span analysis to be automated. Sections 2 and 3 describe problems with implementing our melodic-morphing algorithm and time-span analysis. In Section 4 we present our proposed method for the solving the above-mentioned problems. The experiments in Section 5, we show that melodic morphing and time-span analysis can be automating by prioritizing the branches of the time-span tree. We conclude in Section 6 with a brief summary and mention of future work.

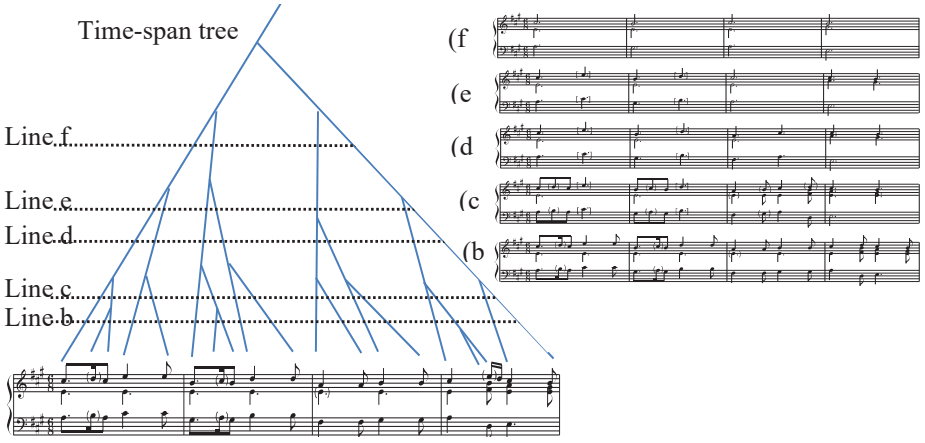


Fig. 1. Time-span tree and melody reduction

2 Implementation Problems of Melodic-Morphing Algorithm

The meaning of morphing is to change something, such as an image, into another through a seamless transition. For example, a method of morphing one face picture into another creates intermediate pictures through the following operations.
(a1) Link characteristic points such as eyes and nose, in the two pictures (Fig. 2a).
(a2) Rate the intensities of the shape (position), color, etc. in each picture.
(a3) Combine the pictures.

2.1 Ideas of Melodic Morphing

Similarly, our melodic-morphing algorithm creates intermediate melo-dies with the following operations.
(b1) Link the common pitch events of the time-span trees of two melodies (Fig. 2b).

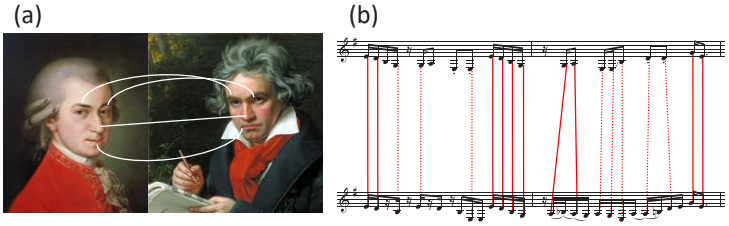


Fig. 2. Examples of linking two pictures/melodies

(b2) Remove those notes that do not reside in the common part by using partial melody reduction, which is explained in the next subsection.
(b3) Combine both melodies.

By using the time-span trees σ_A and σ_B from melodies A and B, respectively, we can calculate the common events of $\sigma_A \sqcap \sigma_B$, which includes not only the essential parts of melody A but also those of melody B (Fig. 3 (b1)). The *meet* operation $\sigma_A \sqcap \sigma_B$ is abstracted from σ_A and σ_B, and those abstracted notes that are not included in $\sigma_A \sqcap \sigma_B$ are regarded to be the difference between σ_A and σ_B.

2.2 Partial Melody Reduction

Music features contained in σ_A and σ_B should exist even in what is not included in the common part. To retrieve these characteristics, we need a method of smoothly increasing or decreasing the number of features. Partial melody reduction abstracts the notes of a melody by using reduction.

With partial melody reduction, we can first acquire melodies α_i (i = 1, 2, ···, n) from σ_A and $\sigma_A \sqcap \sigma_B$ with the following algorithm. The subscript i of α_i indicates the number of notes that are included in σ_A but not in $\sigma_A \sqcap \sigma_B$.

Step 1: Determine the level of abstraction The user determines the parameter L that determines the level of melody abstraction. Parameter L is from 1 to the number of notes that are included in σ_A but not included in $\sigma_A \sqcap \sigma_B$.

Step 2: Abstraction of notes This step involves selecting and abstracting a note that has the fewest dots, obtained from metrical analysis, in the difference of σ_A and σ_B. The numbers of dots can be acquired from the analysis results. If two or more notes have the fewest dots, we select the first one.

Step 3: Iteration Iterate step 2 L times.

Subsumption relations hold as follows for the time-span trees σ_{α_m} constructed with the above algorithm.

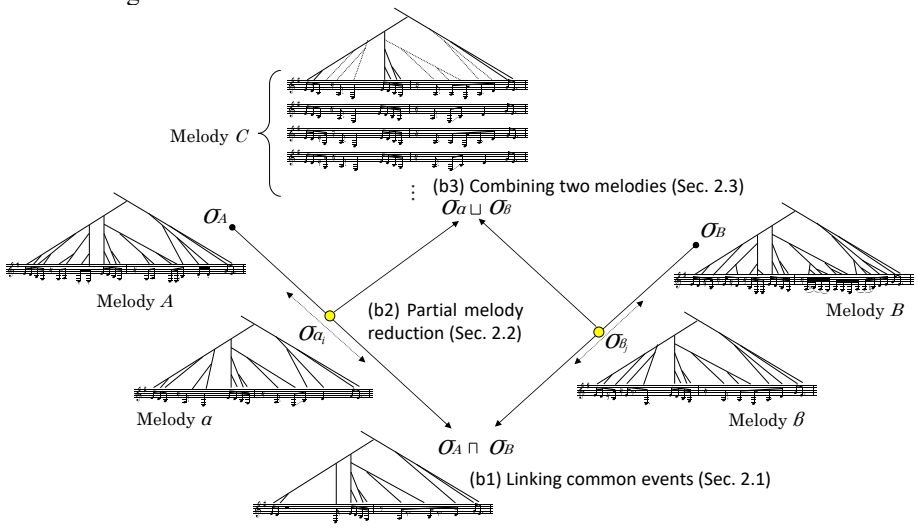


Fig. 3. Overview of melodic-morphing algorithm

$$\sigma_A \sqcap \sigma_B \sqsubsetneq \sigma_{\alpha_n} \sqsubsetneq \sigma_{\alpha_{n-1}} \sqsubsetneq \cdots \sqsubsetneq \sigma_{\alpha_2} \sqsubsetneq \sigma_{\alpha_1} \sqsubsetneq \sigma_A \quad (1)$$

In Fig. 3 (b2), there are nine notes included in σ_A but not included in $\sigma_A \sqcap \sigma_B$. Therefore, the value of n is 8, and we can acquire eight types of melody α_i (i = 1, 2, ···, n) between σ_A and $\sigma_A \sqcap \sigma_B$. Hence, melody αi attenuates features that exist only in melody A.

In the same manner, we can acquire melody β from σ_B and $\sigma_A \sqcap \sigma_B$ as follows.

$$\sigma_A \sqcap \sigma_B \sqsubsetneq \sigma_{\beta_j} \sqsubsetneq \sigma_B \quad (2)$$

2.3 Combining Two Melodies

We use the *join* operator $\sqcup$ to combine melodies σ_{α_i} and σ_{β_j}, which are the results of the partial reduction done using the time-span tree of melodies σ_A and σ_B (Fig. 3 (b3)).

The simple *join* operator is not sufficient for combining σ_{α_i} and σ_{β_j}, because $\sigma_{\alpha_i} \sqcup \sigma_{\beta_j}$ is not always a monophony nevertheless σ_{α_i} and σ_{β_j} are monophonies. In other words, the result of the operation may become polyphony (chords) when the time-span structures overlap and the pitches of the notes differ.

To solve this problem, we introduce a special notation, [n_1, n_2], which indicates note n_1 or note n_2, as a result of $n_1 \sqcup n_2$. Accordingly, the result of $\sigma_\alpha \sqcup \sigma_\beta$ is all possible combinations of monophony.

2.4 Implementation Problems of Melodic-morphing Algorithm

Although we have given priority to automating the morphing process, our melodic-morphing algorithm has the following two problems.

Problem 1: No order of abstract notes. The first problem has to do with the order of abstract notes in partial melody reduction. In Step 2 of Section 2.2, an abstraction is made from the notes with the fewest dots, but this is not always the case, for example, in a time-span tree where there is a structurally salient note on a weak beat. In addition, we have to consider whether it is appropriate to uniquely determine the partial reduction path, as in Equation 1 in Step 3. If there are multiple paths for partial reduction, there is a possibility that more diverse melodies can be output.

Problem 2: Notes with overlapping times occur. The second problem is that the two notes overlapped temporally that may occur in the *join* of two time-span trees. In such cases, it is necessary to manually select one melody from among multiple generated melodies, and it is difficult to completely automate the morphing method. Further, the user remains in the dark as to the morphing process. In particular, it is difficult for the user to understand that the number of melodies output as a result of a number of melodic morphing changes. Even if the user understands the outline of the morphing method in Section 2, the outputs of multiple melodies may not match his or her expectations.

Our approach for automating melodic morphing is to define the order of notes abstracted by partial reduction and the order of notes selected by *join*. That is, when the time-span trees σ_A and σ_B of melodies A and B and the number of notes to be abstracted for each are determined, a unique melody, C, is obtained.

3 Implementation Problems of Deep-learning-based Time-span Tree Analyzer

There are three problems in the deep learning of time-span tree analysis, as follows.

Problem 3: Low Number of Ground Truth Data Sets. As ground truth data of the time-span tree, 300 melodies and their time-span trees are published in the GTTM database [6]. However, the number of data sets (300) is extremely small for learning deep neural networks (DNNs). For a small amount of learning data, over-fitting is inevitable, and an appropriate value cannot be out-put when unknown data are input.

In the time-span analysis by musicologists, the entire time-span tree cannot be acquired at once but gradually analyzed from the bottom up. Therefore, the minimum process of analysis is set as one data set, then the number of data sets is increased. For example, if the DNN [7] directly learns the relationship between a four-note melody and its time-span tree, the number of data sets is only one. If we consider the process of reducing one note to one data set, the number of data sets will be three, as shown in Fig. 4a.

The trained DNN estimates the melody consisting of n-1 notes that is reduced to one note when a melody consisting of n notes is input. A time-span tree for a melody consisting of four notes can be constructed by estimating four to three notes, three to two notes, and two to one note, and combining the results (Fig. 4b).

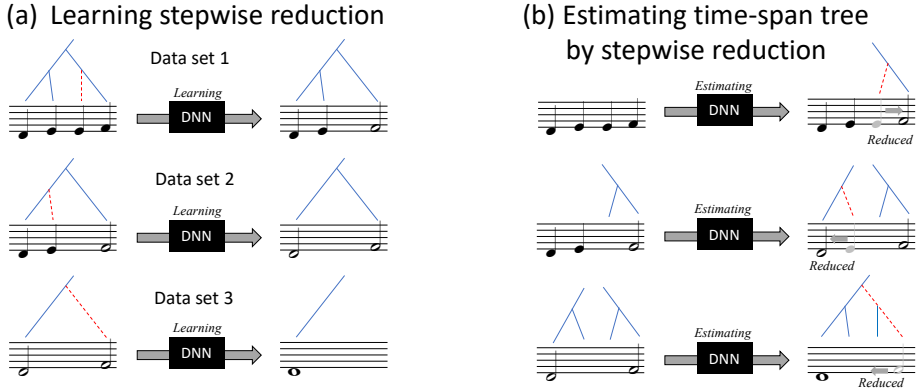


Fig. 4. Learning and estimating by stepwise reduction

Problem 4: Ambiguity of Reduction Process.

Time-span reduction removes decorative notes by pruning from the leaves at the tip of the tree, leaving only structurally important notes in the melody. To implement the stepwise reduction described above, the priority of branches must be obtained in a total order.

However, when it comes to GTTM, there are only a few examples of reduction using a time-span tree, and there is no detailed explanation on the reduction procedure [1]. For example, in Fig. 1, we can see five levels of reduction results, but it is not clear how many levels are necessary.

Marsden *et al*. [8] suggested a means of determining the salience of two note events a and b, neither of which are descendants of the other. They proposed defining the salience of an event as the duration of the maximum of the time spans of the two children at the branching point when the event is generated or where it is reduced.
To automate stepwise reduction, it is more important for the DNN to learn the relationship before and after the reduction than it is to reduce the order of the notes to close to that of human cognition. In Section 4, we propose a time-span tree leveled by the duration of the time span for a simple reduction order that is easy for the DNN to learn.

4 Solution: Time-span Tree Leveled by Duration of Time Span

The *head* in a time-span tree is the top-most pitch event, that is, the most salient in the tree. When two adjacent subtrees are combined, one of the two heads of the subtrees becomes the head of the whole. This indicates that the head of a tree is most salient in the time interval the tree occupies. Since a tree is a hierarchical combination of subtrees, the longest interval of each event in the tree is the most salient as the head of a subtree. Accordingly, we define the base case, when a subtree consists of a single pitch event, to be the duration of the event.

> **Maximum time span:** We call the longest temporal interval when a given pitch event becomes most salient as the maximum time span for the event. In other words, the maximum time span of a pitch event coincides with the temporal duration of the subtree of which the event becomes the head, as a result of the time-span analysis.

The priority of each branch of the time-span tree is determined with a time-span tree drawn with the maximum time span used in the time-span segmentation carried out as the first step of the analysis of time-span reduction. The branch priority is determined in accordance with the following rules.

- Priorities are assigned to each level from the top of the time-span tree drawn with the duration of the time span.
- At the top level, the main branches take precedence.
- At the second and subsequent levels, the higher the priority of a branch X is, the higher the priority of the branch off of X becomes.

Figure 5 shows a time-span tree drawn with the duration of the time span. The branch priority is determined in order from the top in accordance with the first rule. Then, in accordance with the second rule, branch 1 has the highest priority in this time-span tree, and branch 2 has the second-highest priority. The second level in this tree is the double-note level. In accordance with the second rule, the branch off from 1 becomes 3, and that from 2 becomes 4. In the same manner, the priority is determined up to the 16th note level.

4.1 Automatic Melodic Morphing

For automatic partial reduction, we determine how much each melody is to be reduced and reduce the branches of the non-common part of the two melodies. If the

non-common part of the melody of *A* is reduced by 30%, the reduction ratio of melody *B* is determined to be 70%, so that the total is 100%. Then, in the non-common part of each melody, the branches are reduced in order from the branch with the lowest priority. The number of notes is finite, so reducing them in accordance with a set reduction ratio is often impossible. In such cases, the branches are reduced to be closest to the reduction ratio.

As described in Section 2.3, when a melody is synthesized by a *join* operation, the branches of the time-span tree may overlap at the same time. For example, if the branches and notes overlap at the same time due to the *join* operation of melody *A* and *B*, the note with the lower reduction ratio is left. If both reduction ratios are 50%, the note of *A* is left.

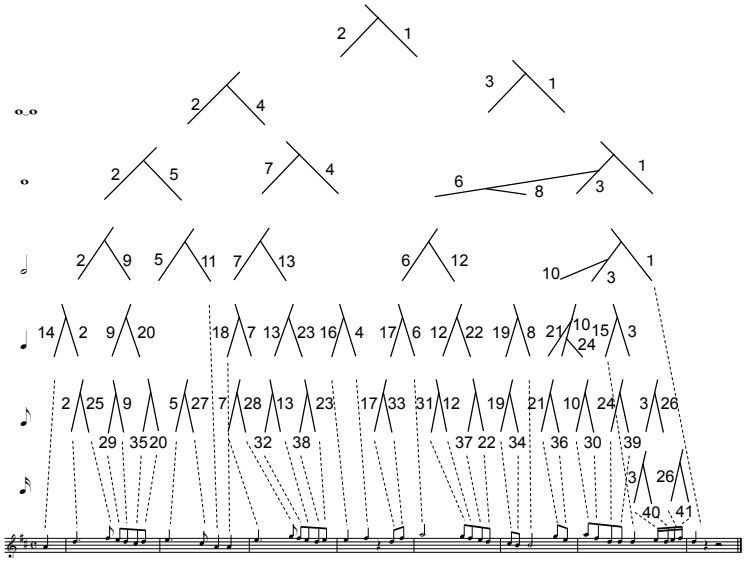


Fig. 5. Time-span tree leveled by duration of time span.

4.2 Automatic Time-span Tree Analysis by Deep Learning

The melody is leveled by the duration of the time span, then it is reduced one note at a time from the lowest level. In the following explanation, when there is a branch, the child branch is called a "sub-branch," and the parent branch is called the "main branch". Since the ground truth data of the time-span tree are mono-phonic, the target is monophony in this paper.

In the time-span tree leveled by the duration of time span, the level of the main branch is always higher than that of a sub-branch. Therefore, if the reduction is carried out in order from the lowest level, the reduction process will proceed without contradiction. It is also important that the reduction process be simple when learning stepwise reduction with a DNN.

Previous time-span tree analyzers (ATTA [9] and sigmaGTTMIII [10]) had low performance because they analyzed in a bottom-up manner using only local information. In contrast, we propose using the entire note sequence before and after stepwise reduction for learning the DNN.

When a recurrent neural network (RNN) [11] or long short-term memory (LSTM) [12] is used as the DNN, the DNN can learn using note sequence, but when a long note sequence is input, the DNN forgets the beginning of it, thus it cannot make use of all the information of the note sequence.

Seq2Seq [13] and Transformer [14] can learn and predict using the information of the entire note sequence. The difference between Seq2Seq and Transformer is the representation of position in the note sequence: Seq2seq uses relative positions by sequentially inputting sequence data into the RNN, while Transformer has an independent additional layer of position information and uses the absolute position.

Therefore, if the absolute position is important for stepwise time-span reduction, Transformer will have high performance, and if the relative position is important, Seq2Seq will have high performance. We evaluated which of the two has the higher performance, as described in Section 5.2.

5 Experiment and Results

As a verification of the usefulness of our proposed time-span tree leveled by the duration of time span, we conducted an experiment to confirm whether melodic morphing and time-span tree analysis can be carried out automatically.

5.1 Automating Melodic Morphing by Prioritization of Branches

After acquiring the time-span tree, there was no arbitrariness in the prioritization of the branches, partial reduction, and combination of melodies. Therefore, when the reduction ratio was determined, the morphed melody could be deterministically obtained. In Figure 6, the notes included in melody *A* are displayed with stems up, and those included in melody *B* are displayed with stems down.

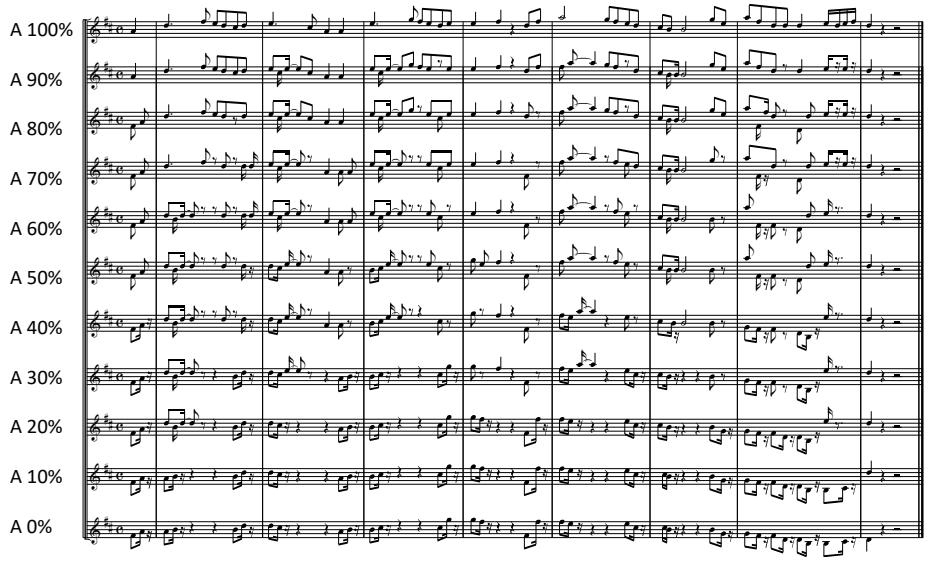


Fig. 6. Results of automatic morphing.

5.2 Comparison of Seq2Seq and Transformer in Stepwise Time-span Reduction

Learning data and evaluation data were created from MusicXML, which are score data, and time-spanXML, which is the ground truth of a time-span tree by the following procedure. The proposed method was first used to reduce (in a stepwise manner) each of the 300 melodies by using the time-span tree, and data before and after the reduction were generated.

Next, the notes in the melodies were made into a one-character string with the pitch and duration concatenated. The pitch was represented as 12 types: C, C#, D, D#, E, F, F#, G, G#, A, A#, and B (excluding octave information). A key that was a major or minor key was then changed to C major or A minor. The duration was represented by multiplying the duration elements of MusicXML by 4. By multi-plying by 4, the duration of most notes became an integer, but since there were melodies containing only a few triplets, quintuplets, sextuplets, and septuplets, the duration was rounded up to an integer. Then, a space was inserted between the strings to represent notes. Finally, in the note sequence after the reduction, "r" was inserted at the position of the reduced note so that we would know which note had been reduced.

The Seq2Seq and Transformer models were both trained with 7362 stepwise time-span-reduction training data sets generated from 270 songs from a GTTM database consisting of 300 pieces, and 849 evaluating data sets were generated from the remaining 30 pieces. Table 1 shows the accuracy of matching the evaluation data and prediction data after 20,000 epochs of training. We can see that Transformer outperformed Seq2Seq in stepwise time-span reduction. Learning was carried out using Nvidia Quadro RTX5000 for laptops [15], and the learning time of Seq2Seq was six days, which is much longer than the seven hours taken by Transformer.

Table 1. Comparison of Seq2Seq and Transformer models.

	Seq2Seq	Transformer
Accuracy	0.90	0.99

6 Conclusion

We proposed the introduction of time-span tree leveled by the duration of time span to problems that are difficult to automate due to the lack of prioritization of time-span tree branches. Experimental results confirmed that melodic morphing and time span analysis based on deep learning can be automated.

We plan to develop various applications and content by using a time-span tree. Our morphing method has appeared in the smart-phone applications of Melody Slot Machine [16], which has a huge number of downloads. By using an automated morphing system, it is possible to build a system that facilitates the addition of content on Melody Slot Machine.

References

1. Lerdahl, F. and Jackendoff, R.: *A Generative Theory of Tonal Music*. MIT Press, Cambridge, 1985.
2. Hamanaka, M., Hirata, K., and Tojo, S.: Melody Morphing Method Based on GTTM. In: *Proceedings of International Computer Music Conference* (ICMC2008), pp. 155–158, 2008.
3. Hamanaka, M., Hirata, K., and Tojo, S.: Melody Extrapolation in GTTM Approach. In: *Proceedings of International Computer Music Conference* (ICMC2009), pp. 89–92, 2009.
4. Hamanaka, M., Hirata, K., and Tojo, S.: deepGTTM-I: Local Boundaries Analyzer Based on Deep Learning Technique. In: *Proceedings of International Symposium on Computer Music Multidisciplinary Research* (CMMR2016), pp. 8–20, 2016.
5. Hamanaka, M., Hirata, K., and Tojo, S.: deepGTTM-II: Automatic Generation of Metrical Structure based on Deep Learning Technique. In: *Proceedings of the 13th Sound and Music Conference* (SMC2016), pp. 221–249, 2016.
6. Hamanaka, M., Hirata, K., and Tojo, S.: "Musical Structural Analysis Database Based on GTTM. In: *Proceedings of the 15th International Conference on Music Information Retrieval Conference* (ISMIR2014), pp. 107–112, 2014.
7. Amari, S., Ozeki, T., Karakida, R., Yoshida, Y., and Okada, M. Dynamics of Learning in MLP: Natural Gradient and Singularity Revisited. *Neural Comput.*, vol. 30, issue 1, pp. 1–33, 2018.
8. Marsden, A., Tojo, S., and Hirata, K. No longer 'somewhat arbitrary': calculating salience in GTTM-style reduction. In: *Proceedings of the 5th International Conference on Digital Libraries for Musicology* (DLfM'18), pp. 26–33, 2018.
9. Hamanaka, M., Hirata, K., and Tojo, S.: ATTA: Automatic Time-Span Tree Analyzer Based on Extended GTTM. In: *Proceedings of the 6th International Conference on Music Information Retrieval Conference* (ISMIR 2005), pp. 358–365, 2005.
10. Hamanaka, M., Hirata, K., and Tojo, S.: σGTTM III: Learning-Based Time-Span Tree Generator Based on PCFG. In: *Proceedings of International Symposium on Computer Music Multidisciplinary Research* (CMMR2015), pp. 387–404, 2015.
11. Pineda, J. F. Generalization of Back-propagation to Recurrent Neural Networks. *Physical Review Letters*, 19(59): 2229–2232, 1987.
12. Hochreiter, S. and Schmidhuber, J. Long Short-term Memory. Neural *Comp.* 9(8): 1735–1780, 1997.
13. Sutskever, I., Vinyals, O. and Le, V. Q. Sequence to Sequence Learning with Neural Networks. In: *Advances in Neural Information Processing Systems 27*, pp. 3104–3112, 2014.
14. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, N. A., Kaiser, L. and Polosukhin, I. Attention Is All You Need. In: *Proceedings of the 31st Conference on Neural Information Processing Systems* (NIPS2017), vol. 30, pp. 6000–6010, 2017.
15. Nvidia: Quadro for laptops, https://www.nvidia.com/en-us/design-visualization/quadro-in-laptops/, 2021.
16. Hamanaka, M. Melody Slot Machine. https://gttm.jp/hamanaka/en/melodyslotmachine/, 2021

Studying Structural Regularities through Abstraction Trees

Filippo Carnovalini[1,2], Nicholas Harley[2], Steve Homer[2], Antonio Rodà[1], and Geraint A. Wiggins[2,3]

[1] University of Padova, Italy
[2] Vrije Universiteit Brussel, Belgium
[3] Queen Mary University of London, UK
filippo.carnovalini@dei.unipd.it

Abstract. "Structure" is a somewhat elusive concept in music, despite being of extreme importance in a variety of applications. Being inherently a hidden feature, it is not always explicitly considered in algorithms and representations of music. We propose a hierarchical approach to the study of musical structures, that builds upon tree representations of music like Schenkerian analysis, and adds additional layers of abstraction introducing pairwise comparisons between these trees. Finally, these representations can be joined into probabilistic representations of a music corpus. The probability distributions contained in these representation allow us to use concepts from Information Theory to show how the structures we introduce can be applied to musicological and music information retrieval applications.

Keywords: Structure, Schenkerian Analysis, Music Representations

1 Introduction

"Structure" is a term that, even only considering music, can assume a variety of meanings. One common use of this term relates to form: the chaining of different sections to create a longer musical piece where some sections are repeated, with or without variations. Another use is more related to shorter melodic fragments, and relates to how a melody can be divided into periods, phrases, and motifs. In this latter case, a musicologist who wishes to analyze structure will try to divide the music into smaller segments and to find similarities, repetitions, inversions, parallel movements or otherwise links between these segments.

Despite the variety of information that can be gathered by such a process, this kind of analysis is often overlooked in algorithms and representation for computational musicology or music information retrieval. This becomes especially evident when computational systems try to generate novel music after learning some features of music from a given dataset of human compositions [1]. However complex or elegant the model used for the generation, we are still far from obtaining results that are on a par with the starting material. This is generally due to the fact that while these models can capture some aspects of the music they analyze, e.g., typical melodic motifs, they fail to capture the entirety of the hierarchical, structural aspects of music. In many cases, this leads to

algorithms that generate music that sounds reasonable for a short time, but seems to "wander off" as the length of the generated piece increases [2, 3, 5].

In the present work we aim to propose a novel solution to this, using a representation that builds upon existing hierarchical representations of music inspired by Schenkerian Analysis, but that goes further by operating pairwise comparisons between trees reducing small segments of music. These comparisons allow to find the kind of reuse of melodic material that we mentioned before. These structured comparisons are then further abstracted, by considering the comparisons operated on a variety of pieces rather than a single piece. These new representations describe structural regularities within a corpus, and leverage approaches from Information Theory to allow us to isolate more interesting features within the representation and to operate comparisons with other pieces.

1.1 Related Work

This work is linked to a variety of computational musicology applications. Some analysis tools that also abstract tree-like structures based on existing theories of music, such as Schenkerian analysis [12, 13] or GTTM [8, 9], are well known in literature; however, in our proposal the tree representations are not the final goal, but a means of intra-piece comparison, allowing analysis the internal repetition structure. The output is similar to other algorithms meant for form analysis [18], but to our knowledge our approach has never been applied in that field. Finally, Wiggins [19] provides an in-depth theoretical base for the relevance of this approach to music analysis and generation, but does not specify any practical approach to perform the proposed analyses.

2 Representations

The algorithms we describe require the input corpus to be made of monophonic melodies with chord annotations over the melody (lead sheets). We used MusicXML format, but other formats could be appropriate as well.

The first step of the process is to segment the input pieces into segments of equal duration. Depending on the level of detail that is being investigated, a length of one or two measures can be appropriate.

Each segment is then individually analyzed, and from each a tree representing melodic reductions is built, following the algorithm described in [13, 17, 4]. This approach uses a sliding window that passes over the notes in the segment, and every time the window contains two or more notes, one of these is deemed the most important according to the tonality, the metric position, and the current chord. This note is kept and the others are eliminated, and the remaining note is made longer to fill the void left by the other notes. At the end of each iteration a new simplified melody is created, and the window is enlarged. At the end of the last iteration, only one note should be left. By stacking the obtained simplifications, a tree similar to those created by analyses such as the ones contained in GTTM [11] or in Schenkerian Analysis [15] is obtained. For this reason we call this tree *Schenkerian Tree* or simply *Sk_tree*.

Once the Sk_trees are built, it is possible to operate pairwise comparisons between them, comparing their roots and recursively comparing the child nodes. In particular, each node in a Sk_tree either represents a note that is present in the original melody (a leaf node) or a note that was created in the process of iterative simplification described above. In the latter case, this node has two or more children, representing the notes that were present in the previous simplification, one of which is kept and the others are eliminated, and it is possible to consider the musical interval of these children. The comparison between nodes of different Sk_trees depends on the content of this interval. The comparable features of these intervals include difference in number of child nodes, difference in pitch intervals between the children, difference in the direction of the children's intervals, or differences in the way that the Schenkerian reduction was performed (for example if the note that was saved was to the left or to the right of the child interval). Since this new structure is based on differences between different sections, we call it *Difference Tree* or *Diff_tree* Figure 1 shows how a Diff_tree can be built from the comparison of two Sk_trees.

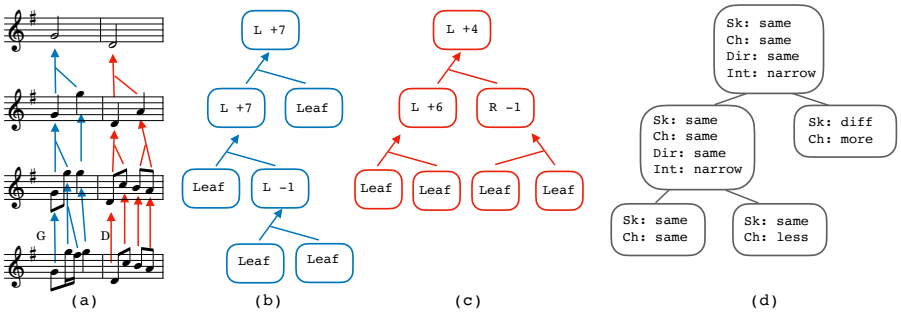


Fig. 1. One example of a simplified diff_tree (d), constructed from two sk_trees (b and c), each representing one of the two measures of the excerpt (a).

For each input piece, a set of $\frac{(n-1)(n)}{2}$ Diff_trees are produced, where n is the number of segments the input piece is divided into. This number is due to the fact that the segments are not compared with themselves nor the segments prior to them, but only with segments that come later in the musical piece, so not all the n^2 possible comparisons are performed. Each Diff_tree is then labelled to indicate which segments are compared in that tree (e.g. if the first and fourth segments are compared, the tree is labeled '0-3'). Once the Diff_trees for a set of pieces are produced, the Diff_trees that share the same label (i.e., that refer to segments in the same position but coming from different pieces) can be joined together into a single tree, that abstracts the general development of that particular comparison. For this reason, we call this representation an *Abstraction Tree* or *Abs_tree*. The procedure works as follows: a new node which will be the root of the Abs_tree is created. Starting from the root of all considered Diff_trees, for each of the possible features, the new node annotates all the possible values that the feature assumes in all the given Diff_trees and the number of occurrences of those val-

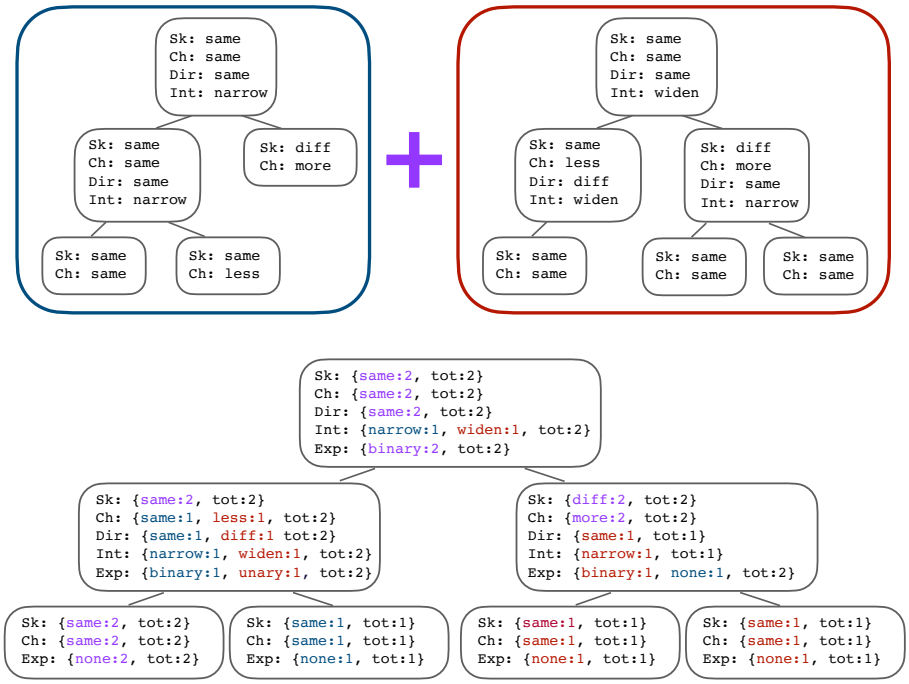


Fig. 2. A simplified Abs_tree built from the two Diff_trees on top. For readability, the tree reports frequencies of occurence and the total number of observation rather than the probabilities that need to be computed with the Bayesian Estimator. The colors represent the tree from which each value for each feature comes: blue for left-side tree, red for the right-side one, and purple for those values that are found in both.

ues. The new node also annotates how many children the roots of the given Diff_trees have, as a new separate feature. Then the algorithm repeats recursively as long as at least one Diff_tree node has at least one child node. Once the recursion process is complete, it is possible to compute the probability of each value v for each feature in each node, using the following Bayes Estimator [16]:

$$p(v) = \frac{occ(v) + \beta}{tot + \beta * size} \tag{1}$$

where $occ(v)$ is the number of occurrences of the value v, while tot and $size$ are respectively the total of samples for that feature and the number of possible distinct values for that feature. β was set to 1. All the features for which tot is less than a certain threshold (we used 5 in the experiments described below) can be removed as those features would entail too little information about the corpus. The nodes that remain empty because of this can be removed as well. Figure 2 shows a simplified example of an Abs_tree built

from two Diff_trees. This process basically creates a probability distribution for each node, consisting of a set of stochastic variables depending on the features present in the Diff_trees. Because of this the Abs_tree is similar to a Markov Model, but rather than having the probability distributions vary in time depending only on the previous state, the only feature that determines the distribution is the position on the tree. For this reason, it would be misleading to think of an Abs_tree as a chain or an automaton, and it is best to only view it as a static probabilistic description of a musical corpus.

3 Applications

In this section, we demonstrate, through example applications, the utility of the abstract representations we introduced. To do so, we will apply the explained structures to tasks relevant to computational musicology and music information retrieval.

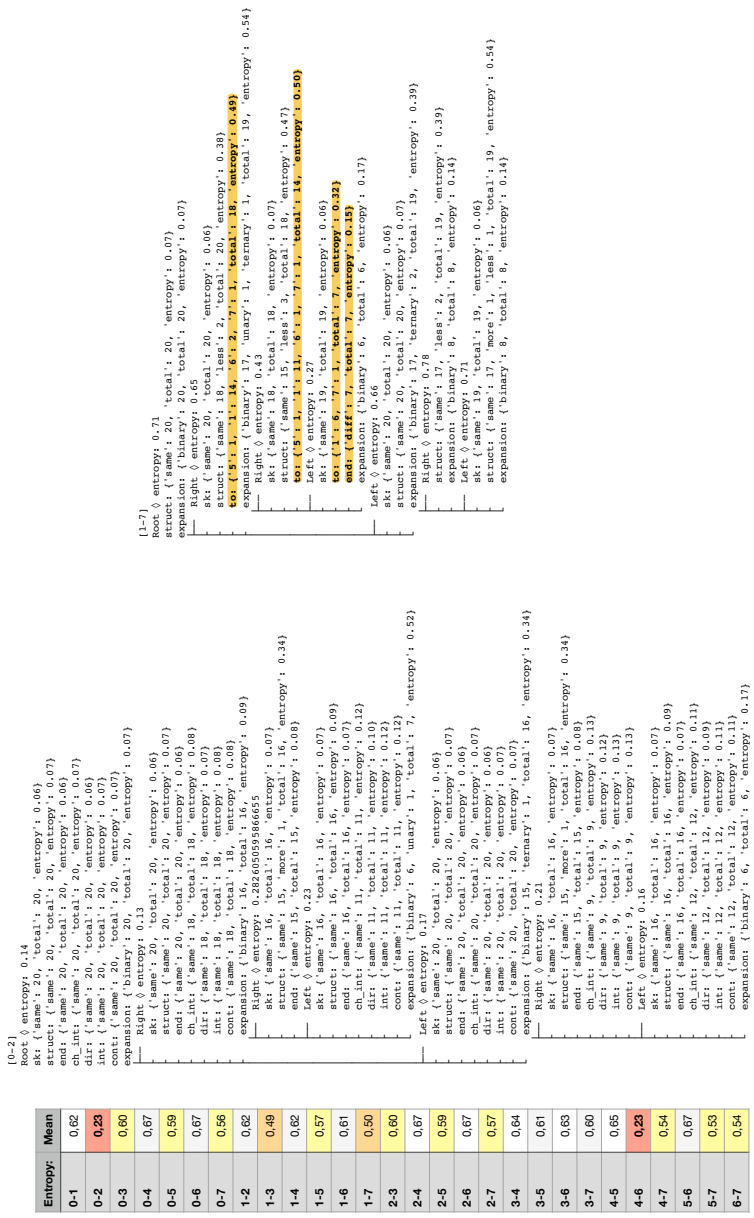
3.1 Regularity Detection within a Corpus

As a first example, we show how the above introduced Abstraction Trees can be inspected to find regularities within a given corpus. As an example corpus, we will use the Leone dataset, a set of twenty-four baroque allemandes [6]. Of these twenty-four we selected the twenty that have sixteen measures in total, to make comparisons easier thanks to the equal length. All the pieces were divided into two-bar segments and the abstraction trees pairwise comparing the segments were built as described above. The procedure produces a large amount of data which is difficult to interpret on its own, but the tools of information theory can help find the most relevant features. For each feature in each node, it is possible to compute the normalized entropy (efficiency) of the probability distribution it describes, which gives a useful indication of the importance of that feature within the corpus. The lower the entropy, the more strictly that feature describes a recurring element in the corpus.

For example, as can be seen in Figure 3, looking at the mean normalized entropy of all the constructed abstraction trees, it becomes evident that the tree comparing segments 0 and 2 (shown in figure) and the one comparing segments 4 and 6 are the most regular ones. In those trees, almost each feature is set to “same”, meaning that there are little or no differences between the above mentioned pairs of segments. Indeed, the phrases in this corpus tend to repeat after 4 measures (the distance between the start of segments 0 and 2), and that is captured by the abstraction tree. Moreover, while the phrases repeat, their ending is varied to make for more definitive phrase endings. This is captured in the abstraction tree comparing segments 1 and 7 (shown in figure) where the left side of the tree shows a repetition like the one described above, but in the right side of the tree the most relevant feature is the one describing the ending grade, which is usually the tonic, as expected from the closing of a musical period.

3.2 Genre Discrimination

The following example uses another metric commonly used in Information Theory. While entropy is related to regularity in a probability distribution, Information Content gives an indication of how unexpected a certain outcome is with respect to a given



Entropy:	Mean
0-1	0,62
0-2	0,23
0-3	0,60
0-4	0,67
0-5	0,59
0-6	0,67
0-7	0,56
1-2	0,62
1-3	0,49
1-4	0,62
1-5	0,57
1-6	0,61
1-7	0,50
2-3	0,60
2-4	0,67
2-5	0,59
2-6	0,67
2-7	0,57
3-4	0,64
3-5	0,61
3-6	0,63
3-7	0,60
4-5	0,65
4-6	0,23
4-7	0,54
5-6	0,67
5-7	0,53
6-7	0,54

Fig. 3. A table summing up the mean entropy of each abstraction tree derived from the corpus, and some examples of abstraction trees as a text output of the software. Only the first three levels were kept for readability. The labels of the tree represent the compared segments: for example ”0-1” means that the first two bars of a piece are compared to measures 3 and 4, since in this case each segment was two measures long.

probability distribution. Since musical cognition is strongly related to expectation [10], this metric becomes a relevant indicator when analyzing musical pieces [14]. In this experiment, we learnt a set of abstraction trees from the 20 allemandes taken from the corpus mentioned above, and the difference trees from a set of 20 reels from the Nottingham Dataset [7], and from 20 jazz pieces composed between 1921 and 1930 taken from the EWLD corpus [17]. The abstraction trees contain probability distributions for each feature in each node, while difference trees can be considered as outcomes for the same features. This means that for each feature it is possible to compute the information content. To give a single measure of the total information content of a difference tree compared to an abstraction tree, the mean of all the features in a node is computed to give the information content of a single node, and the mean of all the information contents across nodes is computed to give the general information content of a tree. This latter mean is also weighted by the mean entropy of the nodes, and by an added coefficient that makes nodes lower in the tree less important than nodes in the upper part of the tree ($depth\_k$ in the formula below). The total formula is described below, where $p(\textit{diff\_tree\_feature})$ represents the probability $p(v)$ (computed according to the estimator 1) of the value v found for the considered feature f in the diff_tree node.

$$\textit{ic}(\textit{tree}) = \frac{\sum_{\textit{node}\in\textit{tree}} \textit{ic}(\textit{node}) \frac{1}{\textit{ent}(\textit{node})} * \textit{depth\_k}(\textit{node})}{\sum_{\textit{node}\in\textit{tree}} \frac{1}{\textit{ent}(\textit{node})} * \textit{depth\_k}(\textit{node})} \tag{2}$$

$$ic(\textit{node}) = \sum_{f\in\textit{node}} \frac{-\log_2(p(\textit{diff\_tree\_feature}))}{\textit{ent}(f)} \tag{3}$$

$$\textit{ent}(\textit{node}) = \frac{\sum_{f\in\textit{node}} \textit{ent}(f)}{\textit{number\_of\_features\_in\_node}} \tag{4}$$

$$\textit{ent}(f) = \sum_{v\in\textit{alphabet}(f)} -\log_2(p(v))p(v) \tag{5}$$

Figure 4 shows the results of the comparisons. Since computing the information content of a piece included in the abstraction tree would be an unfair advantage, similarly to the bias one would get by evaluating a machine learning model using the same dataset that was used for learning the model, an approach similar to a k-fold validation was used. The allemande corpus was split into four parts of 5 pieces, and the information content of each piece was computed with respect to the abstraction trees built solely on the 15 pieces outside the considered allemande's group. This means that there were actually four sets of abstraction trees built each on a different subset of 15 pieces. The values for the other two groups (Jazz and Nottingham) were computed on all the four sets and the mean is reported.

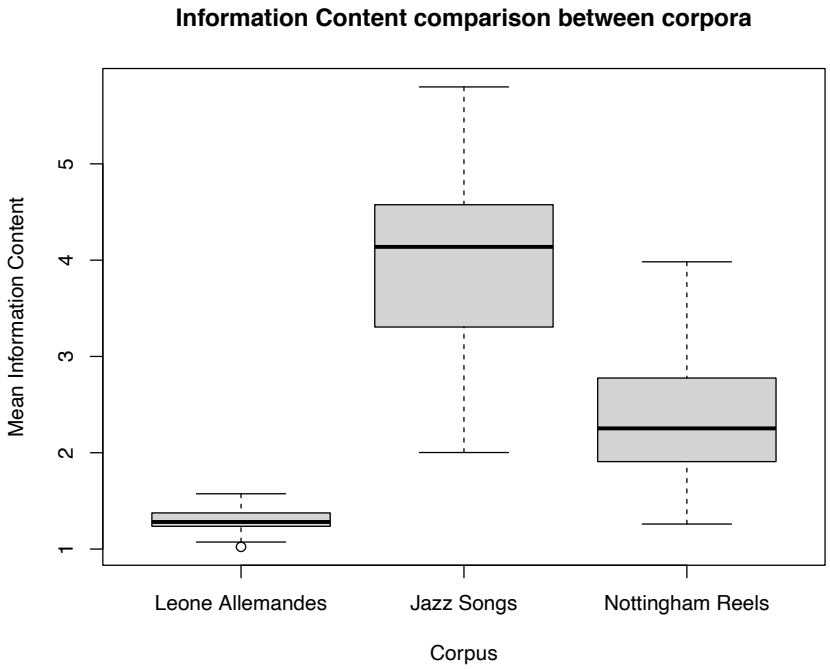


Fig. 4. Comparison of the mean information content computed from each of the three sets of twenty musical pieces. Mean refers to the mean of the trees of a single piece, rather than the mean of an entire corpus.

The results clearly show that this approach is capable of detecting the structural differences between the corpora. The allemandes show a strong structural regularity, that is not found in the other pieces. As expected, the reels from the Nottingham dataset are less unexpected than the jazz pieces, since they too have some structural regularities that are not always found in jazz pieces. It is worth noting that being based on difference trees, what this system captures is the general structure of the piece and how much reuse of melodic material is present, rather than comparing for instance the regularities in the melodies and how typical they are for each genre, possibly making this metric a complementary indicator that could be used in combination with other approaches in genre detection.

4 Conclusions

In this work, we have introduced a novel representation of musical content aimed at encoding in a hierarchical manner features relating to musical structure. The approach builds upon tree-based methods inspired by Schenkerian analysis, but adds additional abstraction layers to describe regularities in a musical corpus rather than in a single

piece. The final representation uses probability distributions, that can be analyzed using tools from Information Theory as we show through two examples in the latter part of the paper. The system as described here is capable of detecting regularities in a simple allemande corpus, but the general approach can be adapted to a variety of specific algorithms: the algorithm used for the construction of Schenkerian trees could be changed, as well as the set of features used to compare them and build the Difference trees, potentially adapting to different kinds of music and different analysis needs. One of the biggest drawbacks of the current implementation is that it is based on a fixed window length, making it harder to capture smaller structural features. An algorithm for segmentation could be embedded in the system to detect the best subdivision of a piece, but the general algorithm would need to be modified to adapt to segments of unequal length.

Applying this approach to other structures opens interesting directions for future works, but even keeping the system as presented now it is possible to further investigate its applications. One of the motivations behind this work was to find descriptions of music structure that can be used when generating computer-composed music. In this scenario, Abstraction trees could offer a useful metric to describe how well a generated musical piece respects the typical structure of a certain style by computing the Information Content in comparison with a goal corpus.

While this work is not meant to give a comprehensive descriptor of all musical aspects of a corpus, we believe that this contribution might help formalizing some aspects of music that are sometimes overlooked in favour of more prominent aspects such as melody, rhythm, and harmony.

Acknowledgments. FC is funded by a doctoral grant by University of Padova and by a mobility grant by Fondazione Ing. Aldo Gini. NH, ST and GW received funding from the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme.

References

1. Briot, J.P., Hadjeres, G., Pachet, F.D.: Deep Learning Techniques for Music Generation. Computational Synthesis and Creative Systems, Springer International Publishing, New York, NY (2020). https://doi.org/10.1007/978-3-319-70163-9, https://www.springer.com/gp/book/9783319701622
2. Briot, J.P., Pachet, F.: Deep learning for music generation: challenges and directions. Neural Computing and Applications **32**(4), 981–993 (Feb 2020). https://doi.org/10.1007/s00521-018-3813-6, https://doi.org/10.1007/s00521-018-3813-6
3. Carnovalini, F.: Open Challenges in Musical Metacreation. In: Proceedings of the 5th EAI International Conference on Smart Objects and Technologies for Social Good. pp. 124–125. ACM, Valencia Spain (Sep 2019). https://doi.org/10.1145/3342428.3342678, http://dl.acm.org/doi/10.1145/3342428.3342678
4. Carnovalini, F., Rodà, A.: A Multilayered Approach to Automatic Music Generation and Expressive Performance. In: 2019 International Workshop on Multilayer Music Representation and Processing (MMRP). pp. 41–48. IEEE, Milano, Italy (Jan 2019). https://doi.org/10.1109/MMRP.2019.00016, https://ieeexplore.ieee.org/document/8665367/

5. Carnovalini, F., Rodà, A.: Computational Creativity and Music Generation Systems: An Introduction to the State of the Art. Frontiers in Artificial Intelligence **3**, 14 (Apr 2020). https://doi.org/10.3389/frai.2020.00014, https://www.frontiersin.org/article/10.3389/frai.2020.00014/full
6. Carnovalini, F., Rodà, A., Harley, N., Homer, S.T., Wiggins, G.A.: A New Corpus for Computational Music Research andA Novel Method for Musical Structure Analysis. In: Audio Mostly 2021 (AM '21). p. 4. ACM, virtual/Trento Italy (2021). https://doi.org/10.1145/3478384.3478402, https://doi.org/10.1145/3478384.3478402
7. Foxley, E.: Nottingham Database (2011), https://ifdo.ca/~seymour/nottingham/nottingham.html
8. Hamanaka, M., Hirata, K., Tojo, S.: Implementing methods for analysing music based on lerdahl and jackendoff's generative theory of tonal music. In: Computational Music Analysis, pp. 221–249. Springer, New York, NY (2016)
9. Hamanaka, M., Hirata, K., Tojo, S.: deepgttm-iii: Multi-task learning with grouping and metrical structures. In: International Symposium on Computer Music Multidisciplinary Research. pp. 238–251. Springer, Matosinhos, Porto (2017)
10. Huron, D.: Sweet Anticipation: Music and the Psychology of Expectation. MIT Press (Jan 2008)
11. Lerdahl, F., Jackendoff, R.S.: A generative theory of tonal music. MIT press, Cambridge, MA (1985)
12. Marsden, A., Hirata, K., Tojo, S.: Towards computable procedures for deriving tree structures in music: Context dependency in GTTM and Schenkerian theory. In: Proceedings of the Sound and Music Computing Conference 2013. pp. 360–367. KTH Royal Institute of Technology, Stockholm, Sweden (2013)
13. Orio, N., Rodà, A.: A measure of melodic similarity based on a graph representation of the music structure. In: ISMIR. pp. 543–548. ISMIR, Kobe, Japan (2009)
14. Pearce, M., Wiggins, G.A.: Expectation in melody: The influence of context and learning. Music Perception **23**, 377–405 (2006)
15. Schenker, H.: Free Composition (Der freie Satz). Longman Music Series, Longman, New York, NY, USA (1979)
16. Schürmann, T., Grassberger, P.: Entropy estimation of symbol sequences. Chaos: An Interdisciplinary Journal of Nonlinear Science **6**(3), 414–427 (Sep 1996). https://doi.org/10.1063/1.166191, http://aip.scitation.org/doi/10.1063/1.166191
17. Simonetta, F., Carnovalini, F., Orio, N., Rodà, A.: Symbolic Music Similarity through a Graph-Based Representation. In: Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion - AM'18. pp. 1–7. ACM Press, Wrexham, United Kingdom (2018). https://doi.org/10.1145/3243274.3243301, http://dl.acm.org/citation.cfm?doid=3243274.3243301
18. Velarde, G., Meredith, D.: A wavelet-based approach to the discovery of themes and sections in monophonic melodies. In: International Symposium on Music Information Retrieval: ISMIR. p. 4. ISMIR, Taipei, Taiwan (2014)
19. Wiggins, G.A.: Structure, abstraction and reference in artificial musical intelligence. In: Handbook of Artificial Intelligence for Music. Springer Nature Switzerland AG, Cham (2021), https://doi.org/10.1007/978-3-030-72116-9_15

Symbolic Textural Features and Melody/Accompaniment Detection in String Quartets

Louis Soum-Fontez[1], Mathieu Giraud[2], Nicolas Guiomard-Kagan[1], and Florence Levé[1,2]

[1] Université de Picardie Jules Verne, MIS, F-80000 Amiens, France
[2] Université de Lille, UMR CNRS 9189 CRIStAL, F-59000 Lille, France
florence.leve@algomus.fr ⋆

Abstract. Music is often described as melody and accompaniment, and several MIR studies try to identify melodies. But the organization of voices is not limited to such a distinction between melody and accompaniment: *Textural effects* – such as repeated notes, syncopes, homorhythmy, parallel moves or imitation – underline the melody/accompaniment layout, and changes in texture usually mark structural transitions in music. We investigate how textural and other characteristics can help to identify melodic voices in polyphonic music. We select *measure-level features* to analyze symbolic scores of string quartets, including new *textural features*, and propose models to predict, on each measure, *melodic and accompaniment layers* in such scores, each layer possibly including several instruments. We evaluate these sets of features and the models on 12 movements in Haydn and Mozart string quartets. The best models have an average accuracy of more than 85%, taking into account both statistical and textural features.

1 Introduction

Melody, as the foreground of a musical material, is complex to define and characterize. In *string quartets*, the first violin (Vln1), as a leading instrument, often plays the main melody. However, the three other instruments of the quartet – second violin, viola, and cello – can also join the melody or play it alone over time (Figure 1).

Melody Detection. Research on melody extraction is an active field in the audio domain [23, 8, 14]. In the symbolic domain, studies investigated the melodic content of monophonic phrases and patterns through the lens of melodic similarity [31], melodic segmentation [1, 19, 29, 30], and contour analysis [25, 24].

Concerning the particular question of identifying the melody in a polyphonic score, Uitdenbogerd and Zobel [28] proposed several algorithms identifying the melodic line in polyphonic MIDI files, including the simple *skyline algorithm* that labels as melody the highest pitch at each onset. Rizo et al. proposed a set of statistical descriptors extracted from each track of MIDI files of different music styles (classical, jazz and pop) and trained a random forest classifier to identify melody tracks in these pieces [22].

⋆ This work is partially funded by French CPER MAuVE (Région Hauts-de-France).

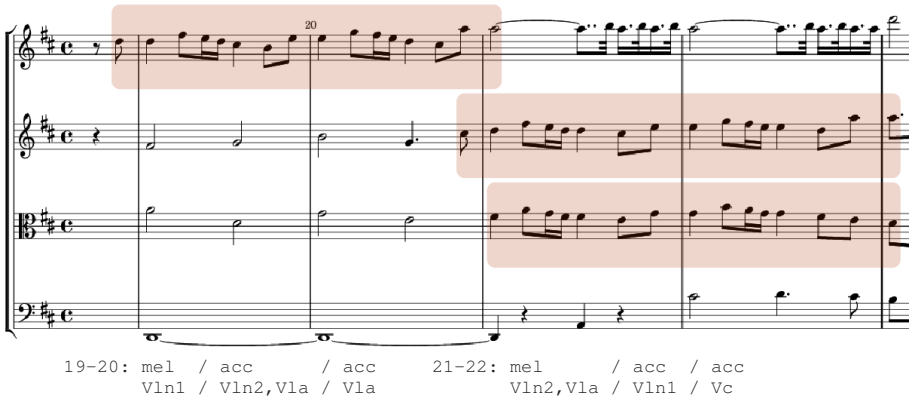


Fig. 1. Haydn, Quatuor op. 33/1, I, mes. 19-22. The texture is described for each measure – even if the melodies are not stricty aligned on measures boundaries. On mesures 21-22, the role of the first violin may be debated, but the main melody is played on the second violin and on the viola, mostly in parallel move in sixths.

Madsen et al. proposed an algorithm for predicting melody notes at any point of the piece, based on a sliding window rendering the complexity of the musical lines [17]. One limitation is that they assume that there is only one melodic line at a time and they reported that the skyline algorithm was still better performing on one Haydn string quartet. They also use this method to identify the melody track in two datasets of popular music [16]. These first results show that assessing complexity may help the recognition of the melody. Friberg et al. tried to recognize the main melody in a polyphonic symbolic score on ringtones of popular music [7], using Huron's perceptual principles [11]. Some of the features they use are derived from symbolic data but intend to model audio features, such as timbre, staccato/legato, or sound level.

Texture and Melody. The role of *texture* has long been recognized in music theories [15, 13], but systematic, formalized, or computed analyses of texture remain few. In 1960, Nordgren quantified some aspects of the orchestral texture [20]. In 1982, Rahn, discussing the melody identification in polyphonies, argues that *a melody "stands out" from its accompanying parts largely on the basis of its complexity* [21]. In 1989, Huron discussed the semantics of the term "texture" and proposed measures to evaluate the textural diversity of music [10].

Several studies focused on the segregation of polyphonic music voices or *streams*, as a listener might perceive them [2, 3, 26]. Duane's thesis [6] further proposed to characterize texture in string quartets by grouping the notes in streams perceived by listeners and by characterizing the *role* of these streams. He described three roles: main lines (including melodies), secondary lines and accompaniment. He established by statistical methods that the perception of textural flows was mainly related to note synchronicity, coordinated pitch modulation (especially parallel movements), as well as the presence

of certain harmonic intervals (metricity, rhythmic repetition, rhythmic patterns, melodic contour, and harmony do not seem to have a significant role).

We previously proposed to describe texture with *layers*, each one determined according to its role and qualified according to its composition [9]. At the first level, layers are mainly qualified as melody, accompaniment, or other minor roles. One melodic layer may include several instruments, and there can be two melodic layers. At the second level, layers can be tagged as repeated notes, syncopation, sustained notes, imitation, and homorhythmy, which can be refined by a complementary descriptor in the case of parallelism, unison, or octave.

Outline. Several textures are more specifically used for rhythmic or accompaniment parts (as repetitions, homorhythmies, or syncopes) or are indicators of some relationship between two voices or more. However, no MIR studies have linked such textures to the analysis of melodic and accompaniment layers. Our goal here is to improve the analysis of textures in symbolic scores, notably the melody/accompaniment detection, focusing on *string quartets*, where melody is often taken by other instruments than the first violin. We aim to improve the understanding of interactions between instruments and the changes in texture by determining melodic and accompaniment layers more precisely. We select measure-level features to analyze symbolic scores of string quartets, gathering existing features [22], and new *textural features* (Section 2). We introduce models to predict melodic and accompaniment layers based on such features (Section 3). We evaluate these sets of features and the models on a set of 12 movements in Haydn and Mozart string quartets and discuss these results (Section 4).

2 Measure-level Features for Melody Detection

To predict whether a measure is melody or accompaniment, we use the following set of features computed on each measure.

2.1 Voice Name (4)

These features enable the (baseline) skyline algorithm, considering that the top voice is the melody.

– (voice-name): 4 binary features, activated depending on the voice (first violin, second violin, viola, cello)

2.2 Statistical Features (20)

The features introduced by Rizo et. al were used to predict, *on the whole piece*, which track is the melody between all tracks [22]. They are linked to music properties that can make what is a melody or what is an accompaniment (see Section 4.2). We computed here these features *on each measure*. They are grouped in 5 categories: track information (normalized duration, number of notes, occupation rate, polyphony rate),

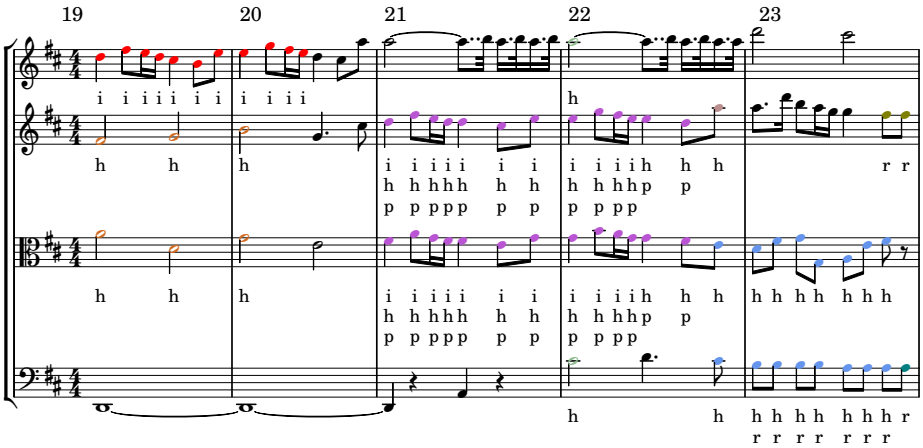


Fig. 2. Detection of individual textures in measures 19-23 of String Quartet op. 33-1, 1st movement by Haydn (see also Figure 1). The colors show the related voices for textures i/h/p.

pitch (highest, lowest, mean, standard deviation), pitch intervals (number of different intervals, largest, smallest, mean, standard deviation), note durations (longest, shortest, mean, standard deviation), and syncopation (number of syncopated notes). We also added the number of repeated notes.

2.3 Textural Features ($7 \times 16 + 1$)

To further describe the music texture, we propose a new set of high-level features describing the organization of notes and voices. The taxonomy of [9] introduced several textures but proposed an algorithm only for homorhythmic layers. Inspired by this taxonomy, we design here the following binary features, that can be computed *on every note*:

- repeated notes (r): We consider as repeated notes sequences of at least three successives notes of the same pitch and the same duration, for a total duration of at least one beat and a half, possibly spaced with rests of at most one beat.
- syncopes (s): A note is considered as a syncope if it starts on a weak beat or on a second half of any beat and continues on at least the next beat.
- homorhythmy (h): Two voices are considered as homorhythmic when they play only notes starting and ending at the same time during at least three beats.
- parallel moves (p): Two homorhythmic voices are considered in a parallel move when at least three close pairs of notes have the same diatonic interval – generally thirds, sixths, or octaves or unison.
- imitation (i): We consider as imitation the repetition of a pattern on some voice, called the original pattern, by another voice with some delay. This can be seen as a parallel move delayed in time. This is computed by a simplification of the Mongeau-Sankoff algorithm [18] requiring here five exact notes or more approximate matches.

– rest (rest): There is no note, but a rest.
– The feature (none) is added when none of the previous six features is activated with the given sixteenth.

On a given measure, these 7 features are actually computed on *each of the 16 sixteenth notes* – the considered corpus being in 4/4, see next section. For each sixteenth note, a vector gives the features that are activated, taking into account an expansion rule – that is including notes that are still sounding but not attacked there. Moreover, the following summarizing feature is added on each measure:

– texture ratio: number of sixteenths on which at least one of the textures is activated, represented as a ratio between 0 and 1

Figure 2 shows an example of the heuristic detection of these textures. The prediction of repeated notes (r) (such as some eights in measure 23) and homorhythmy (h) is very reliable and would be here close to a manual annotation. The parallel move (p) is correctly identified at measures 21-22. The imitation pattern (i) at measures 19-20 on the first violin, that is later taken on the second violin and viola at measure 21-22, is also correctly detected. However, a manual annotation would probably not set the same boundaries for such annotations, for example by ending the homorythmy one note later on the measure 20.

3 Learning Models: Melody/Accompaniment Prediction as a Measure Classification Task

We see the melody/accompaniment prediction as a binary classification problem, given the features presented in the previous section. We choose the measure granularity to be consistent with the reference annotations. The statistical features were normalized into a gaussian distribution (0 ± 1) and almost all the textural features are binary. Theses vectors are gathered into a vector of maximal size $4 + 20 + (7 \times 16 + 1)$ for each measure, and a given melody or accompaniment class for the reference annotation.

3.1 Model Architecture

Two models were tested:

– A random forest (RF) classifier as used by [22], taking the average of a set of 200 decision trees trained on random subsets of features, where data are weighted to account for the unbalanceness of categories.
– A simple neural network (NN) with an initial dropout layers with a rate of 0.5 to reduce overfitting [27], 2 hidden fully connected linear layers (64 then 32), separated by *relu* activation layers and batch normalization layers, and a last layer, composed of a unique neuron, with a *sigmoid* activation function and a threshold of 0.5 between melody/accompaniment prediction. Weights were initialized uniformly. Batch size is 32 and the learning rate is 10^{-3}, with early stopping after 50 iterations without improvement. To estimate errors at each iteration, the loss function used is binary cross-entropy. The optimization of the gradients is done with Adam [12].

Haydn			m	% mel	% Vln1 mel
17.1	i	E Major	111	29.1	97.1
17.2	i	F Major	101	30.9	81.6
17.3	iv	Eb Major	70	34.1	73.1
33.1	i	B minor	91	31.2	72.6
33.2	i	Eb Major	90	34.9	93.5
33.3	i	C Major	165	32.2	100.0
33.4	i	Bb Major	83	35.7	98.3
50.1	i	Bb Major	164	31.4	76.2

Mozart			m	% mel	% Vln1 mel
No. 2, K. 155	i	D Major	119	37.0	96.5
No. 4, K. 157	i	C Major	126	45.6	80.8
No. 6, K. 159	i	Bb Major	71	53.2	98.2
No. 14 K. 387	i	G Major	171	29.5	86.7

Table 1. The corpus contains 12 movements of Haydn and Mozart string quartets, all in 4/4. The last three columns give the number of measures (m), the ratio of measures (on the 4 voices) labeled as melody, and the ratio of measures on the first violin labeled as melody.

4 Results

4.1 Corpus, Implementation, and Availability

The corpus includes 12 movements of string quartets by Haydn and Mozart (Table 1), totaling 1362 measures, as `.krn` files. We extended the corpus of our previous study [9], and we distribute the complete set of annotations as open data at `www.algomus.fr/data`. Each measure on each of the four instruments was labeled as *melody*, *accompaniment*, or *other*. Only measures with *melody* and *accompaniment* were taken into account, totaling 4791 measures. The files were processed with music21 [5], using actual pitch spelling (for example to compute intervals), and the learning models were implemented with keras [4]. The code is available at `www.algomus.fr/code`.

4.2 Statistics on the Features

Figure 3 shows the distribution of some features over the measures labeled as melody or accompaniment in the corpus.

As expected, the features on the pitch (highest, mean, and lowest), being very similar to (voice-name), are very significant to tell apart the melodic and accompaniment parts. More interestingly, other features play a significant role, such as (num-notes) (melody tends to have more notes) or (num-diff-int) (melody tends to use conjoint intervals). In textural features, imitation and syncopes are significantly associated to melody, whereas repeated notes are significantly associated with accompaniment. Homorhythmy and its subset parallel moves are found in both roles, but parallel moves are more used for melody.

4.3 Accuracy over a Leave-one-piece-out Strategy

We did not split these 4791 measures into a training set and a validation/test set: Indeed, having different measures of the same piece in different sets would bring overfitting due to repeats or similar sections inside each piece. Considering the relatively small size of

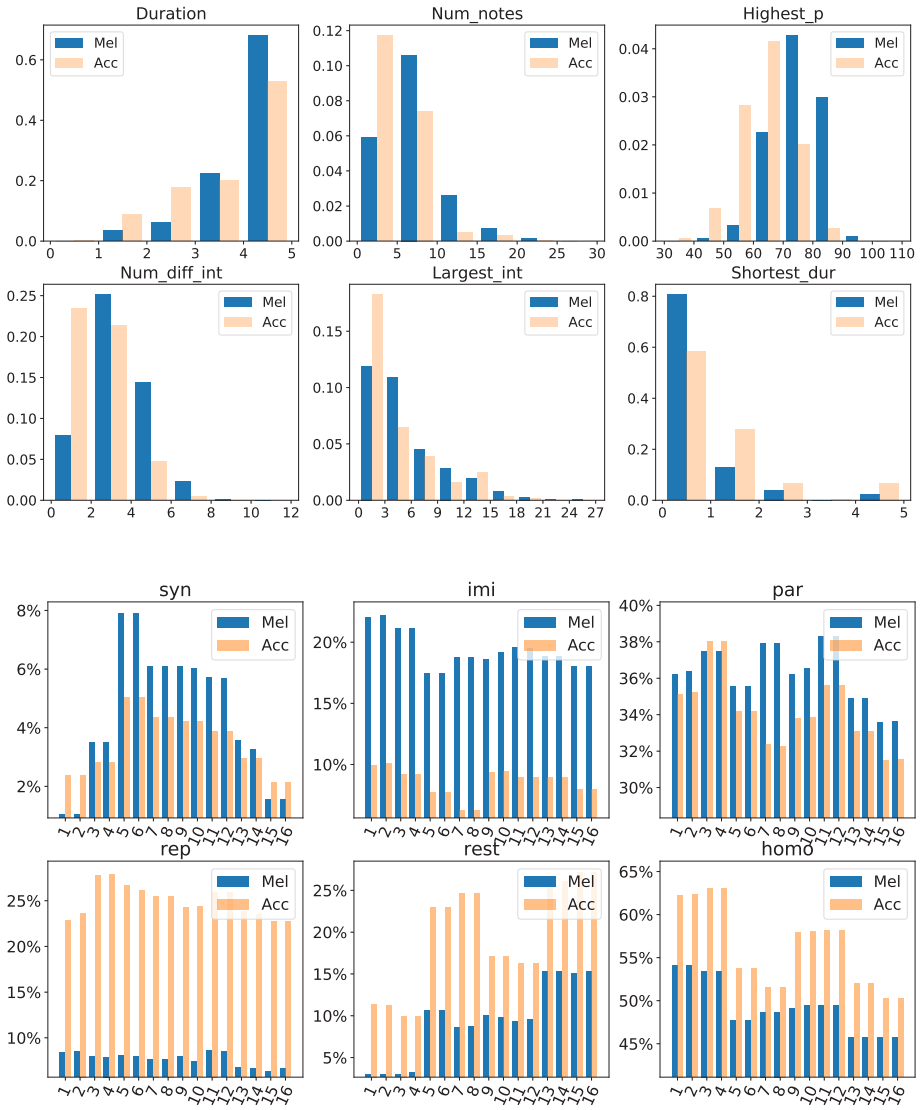


Fig. 3. Distribution of some features in the 12 movements of the corpus, split into the measures labeled as melody (Mel) or accompaniement (Acc) in the reference analysis. The data was normalized in order that each of the area equals to 1. Top two lines: Some of the statistical features introduced by [22]. Bottom two lines: Textural features on each sixteenth on each measure, temporal barplot representing the proportions of a given sixteenth having one of these textures.

	base		RF	NN	
Features	all	dm	all	all	dm
Majority	65.4	19.0	–	–	–
Top (Vln1)	**84.1**	0.0	–	–	–
Statistics	–	–	78.9	81.9	51.9
Texture	–	–	69.1	72.3	**52.8**
Statistics + Texture	–	–	80.3	**82.4**	49.9
V. Name	–	–	84.3	84.2	0.0
V. Name + Texture	–	–	83.1	**86.8**	26.6
V. Name + Statistics	–	–	83.3	84.1	25.4
V. Name + Statistics + Texture	–	–	84.3	85.3	**32.0**

Table 2. Mean accuracy of the baseline models (base), as well of the Random Forest (RF) and the neural network (MLP) on the leave-one-piece-out strategy and various sets of features, evaluated on *all* measures but also on *difficult* measures (dm) i.e. where the first violin is not playing the melody or another voice is playing it.

the corpus, we rather opted for a *leave-one-piece-out* strategy: Each piece is separately considered as a validation set of a model trained on all other pieces. We iterate and report the average accuracy over all the pieces.

As baseline models, we consider *Majority* (all measures are predicted as accompaniment) and *Top* (the first violin, as the top voice, is predicted as the melody – this is equivalent than only considering the (voice-name) feature). Table 2 shows that the best model is the NN taking into account the voice names and our proposed textural features, with about 86.8% of correct predictions. As expected, the (voice-name) alone has significant results – and this is confirmed by the baseline Top(Vln1), 84.1%. The statistical features, even alone, have a good performance, but include features on pitch that are very similar to (voice-name). Conversely, the textural features, without any feature related to pitch, manage alone to identify 72.3% of the measures, including 52.8% on difficult measures where the first violin is the voice playing the melody. Adding these textural features to (voice-name) improves the accuracy.

We call *difficult* measures the 15.8% measures where melody is not at the first violin or shared between several instruments. The best model here correctly predicts the melody in 32% of such measures.

4.4 Focus on Specific Cases

With the best model, the best results are on the first movement of Haydn 33.4 (96.1%), this movement having melodies almost always played on the first violin (see Table 1).

Conversely, Figure 4 details the prediction on five difficult measures in a Mozart quartet. On measures 27-29, the three voices are predicted as accompaniment, whereas the reference annotation labels as melody the second violin. Although the melodic patterns are about the same in measures 27, 30, and 31, it is worth noting that the prediction on measure 27 is wrong, whereas on the measures 30 and 31, the model correctly predicts the melody in parallel moves (p) between one of the violins and the viola: The textural information here helps in predicting the melody.

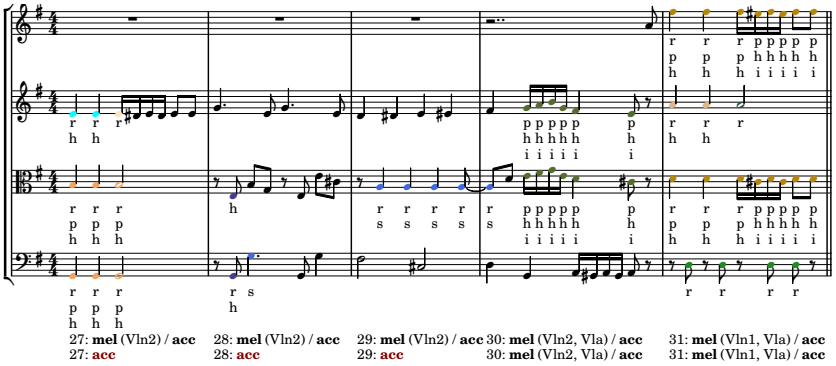


Fig. 4. Textural features, reference annotation (top), and mel/acc prediction by the model NN (bottom) on measures 27 to 31 from quatuor K387 by Mozart, first movement.

5 Conclusion and Perspectives

We evaluated sets of features and models to predict, on each measure, which instrument(s) is playing a melodic content. Experiments on string quartets by Haydn and Mozart show that some textural features are distributed differently in melodic and accompaniment parts, and that the best models detect some of the melodies beyond the first violin or distributed among several instruments. This brings a new step towards a general characterization of melody and texture in polyphonic pieces. Further studies could improve the features and the learning model, generalize such approaches to more complex polyphonic works such as orchestral music, and study the correlation of texture with other parameters such as harmony or form.

References

1. Emilios Cambouropoulos. Musical parallelism and melodic segmentation. *Music Perception*, 23(3):249–268, 2006.
2. Emilios Cambouropoulos. Voice and stream: Perceptual and computational modeling of voice separation. *Music Perception*, 26(1):75–94, 09 2008.
3. Elaine Chew and Xiaodan Wu. Separating voices in polyphonic music: A contig mapping approach. In *International Symposium on Computer Music Modeling and Retrieval (CMMR 2005)*, pages 1–20, 2005.
4. Francois Chollet. *Deep learning with Python*. Simon and Schuster, 2017.
5. Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 637–642, 2010.
6. Ben Duane. *Texture in Eighteenth- and Early Nineteenth-Century String-Quartet Expositions*. PhD thesis, Northwestern University, 2012.
7. Anders Friberg and Sven Ahlbäck. Recognition of the main melody in a polyphonic symbolic score using perceptual knowledge. *Journal of New Music Research*, 38(2):155–169, 2009.
8. Klaus Frieler et al., Don't hide in the frames: Note-and pattern-based evaluation of automated melody extraction algorithms. In *Digital Libraries for Musicology (DLfM 2019)*, pages 25–32, 2019.

9. Mathieu Giraud, Florence Levé, Florent Mercier, Marc Rigaudière, and Donatien Thorez. Towards modeling texture in symbolic data. In *International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 59–64, 2014.
10. David Huron. Characterizing musical textures. In *International Computer Music Conference (ICMC 1989)*, pages 131–134, 1989.
11. David Huron. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception*, 19(1):1–64, 2001.
12. Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR 2015)*, 2015.
13. Katalin Komlós. Haydn's keyboard trios Hob. XV: 5-17: Interaction between texture and form. *Studia Musicologica Academiae Scientiarum Hungaricae*, 28(1/4):351–400, 1986.
14. Ranjeet Kumar, Anupam Biswas, and Pinki Roy. Melody extraction from music: A comprehensive study. In *Applications of Machine Learning*, pages 141–155. Springer, 2020.
15. Janet M. Levy. Texture as a sign in classic and early romantic music. *Journal of the American Musicological Society*, 35(3):482–531, 1982.
16. Søren Tjagvad Madsen and Gerhard Widmer. A complexity-based approach to melody track identification in MIDI files. In *Artificial Intelligence and Music (IWAIM 2007)*, 2007.
17. Søren Tjagvad Madsen and Gerhard Widmer. Towards a computational model of melody identification in polyphonic music. In *International Joint Conference on Artifical Intelligence (IJCAI 07)*, page 459–464, 2007.
18. Marcel Mongeau and David Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.
19. Daniel Muellensiefen, Marcus Pearce, and Geraint Wiggins. A comparison of statistical and rule-based models of melodic segmentation. In *International Conference on Music Information Retrieval (ISMIR 2008)*, pages 89–94, 2008.
20. Quentin R. Nordgren. A measure of textural patterns and strengths. *Journal of Music Theory*, 4(1):19–31, 1960.
21. Jay Rahn. Where is the melody? *In Theory Only: Journal of the Michigan Music Theory Society*, 6:3–19, 01 1982.
22. David Rizo, Pedro J. Ponce De León, Antonio Pertusa, and Jose M. Iñesta. Melody track identification in music symbolic files. In *International Florida Artificial Intelligence Research Society Conference (FLAIRS 2006)*, 2006.
23. Justin Salamon. *Melody Extraction from Polyphonic Music Signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2013.
24. Marcos Sampaio. Contour similarity algorithms. *MusMat*, 2(2):58–78, 2018.
25. Mark Schmuckler. *Tonality and Contour in Melodic Processing*, pages 143–165. Oxford University Press, 2016.
26. Federico Simonetta, Carlos Cancino-Chacón, Stavros Ntalampiras, and Gerhard Widmer. A convolutional approach to melody line identification in symbolic scores. In *International Society for Music Information Retrieval Conference (ISMIR 2019)*, pages 924–931, 2019.
27. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
28. Alexandra Uitdenbogerd and Justin Zobel. Melodic matching techniques for large music databases. In *International Conference on Multimedia (Multimedia 99)*, pages 57–66, 1999.
29. Gissel Velarde, Tillman Weyde, and David Meredith. An approach to melodic segmentation and classification. *Journal of New Music Research*, 42(4):325–345, 2013.
30. Valerio Velardo, Mauro Vallati, and Steven Jan. Symbolic melodic similarity: State of the art and future challenges. *Computer Music Journal*, 40(2):70–83, 2016.
31. Anja Volk and Peter Van Kranenburg. Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*, 16(3):317–339, 2012.

Predominant Instrument Recognition in Polyphonic Music Using Convolutional Recurrent Neural Networks

Lekshmi. C.R and Rajeev Rajan

College of Engineering Trivandrum
APJ Abdul Kalam Technological University
clekshmir04@gmail.com,rajeev@cet.ac.in

Abstract. In this paper, predominant instrument recognition in polyphonic music is addressed using convolutional recurrent neural networks (CRNN) through Mel-spectrogram, modgdgram, and its fusion. Modgdgram, a visual representation is obtained by stacking modified group delay functions of consecutive frames successively. Convolutional neural networks (CNN) learn the distinctive local characteristics from the visual representation and recurrent neural networks (RNN) integrate the extracted features over time and classify the instrument to the group where it belongs. The proposed system is systematically evaluated using the IRMAS dataset. A wave generative adversarial network (WaveGAN) architecture is also employed to generate audio files for data augmentation. We experimented with two CRNN architectures, convolutional long short-term memory (C-LSTM) and convolutional gated recurring unit (C-GRU). The fusion experiment C-GRU reports a micro and macro F1 score of 0.69 and 0.60, respectively. These metrics are 7.81% and 9.09% higher than those obtained by the state-of-the-art Han's model. The architectural choice of CRNN with score-level fusion on Mel-spectro/modgd-gram has merit in recognizing the predominant instrument in polyphonic music.

Keywords: predominant, Mel-spectrogram, modgdgram,convolutional gated recurring unit.

1 Introduction

Predominant instrument recognition refers to the problem where the prominent instrument is identified from a mixture of instruments being played together [16]. In polyphonic music, the interference of simultaneously occurring sounds makes instrument recognition harder. Automatic identification of lead instrument is important since the performance of the source separation can be improved significantly by knowing the type of the instrument [16].

Han *et al.* [16] employed Mel-spectrogram-CNN approach for instrument recognition. Pons *et al.* [22] analyzed the architecture of Han *et al.* in order to formulate an efficient design strategy to capture the relevant information about timbre. Detecting the activity of music instruments using a deep neural network (DNN) through a temporal max-pooling aggregation is addressed in [15]. Dongyan *et al.* [31] employed a network with an auxiliary classification scheme to learn the instrument categories

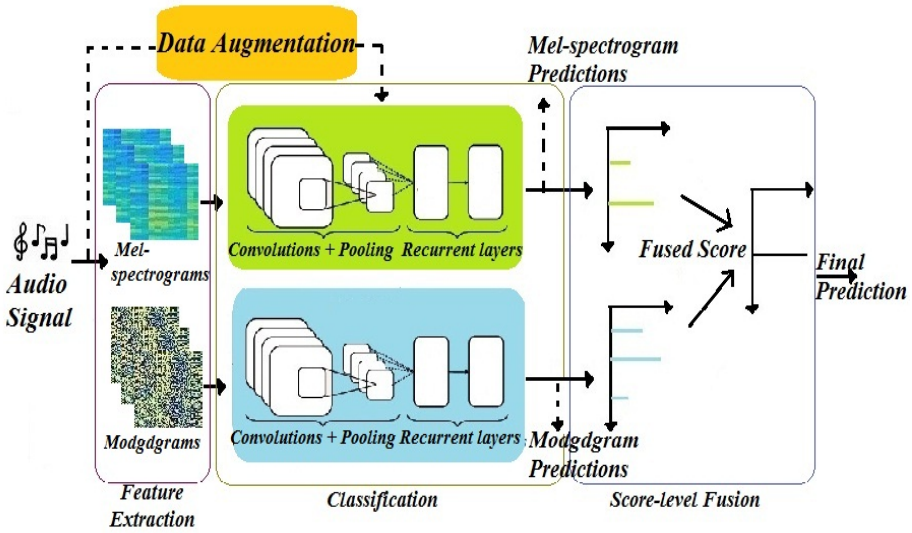


Fig. 1. Block diagram of proposed method of predominant instrument recognition.

through multitasking learning. Gomez *et al.* [14] investigated the role of two source separation algorithms as pre-processing steps to improve the performance in the context of predominant instrument detection tasks. In [18], the Hilbert-Huang transform (HHT) is employed to map one-dimensional audio data into two-dimensional matrix format, followed by CNN to learn the effective features for the task. In [17] an ensemble of VGG-like CNN classifiers is trained on non-augmented, pitch-synchronized, tempo-synchronized, and genre-similar excerpts of IRMAS for the proposed task.

The modified group delay feature (MODGDF) is proposed for pitched musical instrument recognition in an isolated environment in [9]. While the commonly applied mel frequency cepstral coefficients (MFCC) feature is capable of modeling the resonances introduced by the filter of the instrument body, it neglects the spectral characteristics of the vibrating source, which also, play its role in human perception of musical sounds and genre classification [12]. Incorporating phase information is an effective attempt to preserve this neglected component. Some preliminary works on predominant instrument recognition in polyphonic music using group delay functions are discussed in [2, 1]. In [28] a multi-head attention mechanism is employed along with modified group delay functions for proposed task.

In the proposed task, CRNN architecture with score level fusion of Mel-spectrogram and modgdgram is used for recognizing predominant instruments in polyphonic music. Similar approaches combining CNNs and RNNs have been presented recently in many music processing applications [6], [5], [20]. The idea of including modified group delay functions and GAN-based data augmentation strategy are the main contributions of the proposed scheme.

Section 2 explains the system description. Feature extraction is described in Section 3, followed by the model architectures in Section 4. The performance evaluation is

Table 1. Model summary of CNN and CRNN architectures.(* represents the multiplication factor, d_i,f_i,h_i,j_i represents the number of filters used in the networks. (d_i=8,16,24,32,64,128,256,512), (f_i=32,64,128,256), (h_i=8,16,32,64,128,256), (j_i=32,64,128).

*	**Mel-spectrogram-CNN**	**Modgdgram-CNN**	*	**Mel-spectrogram-CRNN**	**Modgdgram-CRNN**
x4	2 X Conv2D (3x3) , d_i	Conv2D (3x3), f_i	x3	2 X Conv2D (3x3) , h_i	Conv2D (3x3), j_i
	Leaky ReLU ($\alpha = 0.33$)	ReLU		Leaky ReLU ($\alpha = 0.33$)	ReLU
	3x3 Max-pooling,stride (3,3)			Batch Normalization	
	Dropout (0.25)			2x2 Max-pooling,stride(2,2)	
	Global Max-pooling			Flatten (1024)	
	Dense (1024)	Dense(512)		2 X Bidirectional LSTM / GRU (32 units)	
	Dropout (0.5)			Flatten (1024)	
	Dense (11), Softmax Activation			Dense (512)	
				Batch Normalization, Dropout (0.5)	
				Dense (11), Softmax Activation	

described in Section 5. The results are analyzed in Section 6. The paper is concluded in Section 7.

2 System Description

The proposed scheme is shown in Fig. 1. In the proposed model, CRNN is used to learn the distinctive characteristics from Mel-spectro/modgd-gram to identify the leading instrument in a polyphonic context. We evaluate the proposed method on the IRMAS dataset and compare its performance to CNN and two variants of RNN-long short-term memory (LSTM) and gated recurring unit (GRU). The performance is also compared with a DNN framework. As a part of data augmentation, additional training files are generated using WaveGAN. During the testing phase, the probability value at the output nodes of the trained model is treated as the score corresponding to the input test file. The input audio file is classified to the node which gives the maximum score during testing. In the fusion framework, the individual scores of Mel-spectro/modgd-gram experiments are fused at the score-level to make a decision. The fusion score S_f, is obtained by,

$$S_f = \beta S_{spectro} + (1 - \beta) S_{modgd} \tag{1}$$

where $S_{spectro}$, S_{modgd}, β are the Mel-spectrogram score, modgdgram score and weighting constant, respectively. The value of β has been empirically chosen to be 0.5. Each phase is explained in detail in the following sections.

3 Feature Extraction

Mel-spectrogram and modgdgram are the inputs used in the proposed scheme. Mel-spectrogram approximates how the human auditory system works and can be seen as the spectrogram smoothed, with high precision in the low frequencies and low precision in the high frequencies [21]. It is computed with a frame size of 50 ms and a hop size of 10 ms with 128 bins for the given task.

Group delay features are being employed in numerous speech and music processing applications [24, 26, 23, 25]. The group delay function is defined as the negative derivative of the unwrapped Fourier transform phase with respect to frequency. Modified group delay functions (MODGD), $\tau_m(e^{j\omega})$ are obtained by,

$$\tau_m(e^{j\omega}) = (\frac{\tau_c(e^{j\omega})}{|\tau_c(e^{j\omega})|})(|\tau_c(e^{j\omega})|)^a, \tag{2}$$

where,

$$\tau_c(e^{j\omega}) = \frac{X_R(e^{j\omega})Y_R(e^{j\omega}) + Y_I(e^{j\omega})X_I(e^{j\omega})}{|S(e^{j\omega})|^{2b}}. \tag{3}$$

The subscripts R and I denote the real and imaginary parts, respectively. $X(e^{j\omega})$, $Y(e^{j\omega})$ and $S(e^{j\omega})$ are the Fourier transforms of signal, $x[n]$, n.$x[n]$ ((weighted signal with index), and the cepstrally smoothed version of $X(e^{j\omega})$, respectively. a and b ($0 < a, b \leq 1$) are introduced to control the dynamic range of MODGD [19, 23]. Modgdgram is the visual representation of MODGD with time and frequency in the horizontal and vertical axis, respectively. The amplitude of group delay function at a particular time is represented by the intensity or color in the third dimension. Modgdgrams are computed with a frame size of 50 ms and hop size of 10 ms using a and b values of 0.9 and 0.5 respectively.

4 Model Architectures

CNNs and RNNs are specific instances of the CRNN architecture presented in this section: A CNN is a CRNN with zero recurrent layers, and an RNN is a CRNN with zero convolutional layers. CNN uses a deep architecture similar to [16] with repeated convolution layers followed by max-pooling. The detailed architecture for Mel-spectrogram and modgdgram CNN and CRNN are shown in Table 1.

RNNs are introduced to handle sequence and time-series data and are well suited for various speech and music-related applications [27], [13]. RNN with sophisticated recurrent hidden units like LSTM and GRU is used because such structures are capable of alleviating the vanishing gradient problem. The designed RNN consists of one input layer, two hidden layers which include two LSTM or GRU layers each with 32 nodes, and an output dense layer with eleven nodes for output classes. ReLU activation is used for hidden layers and softmax is used for the output layer.

In order to benefit from both approaches, the two architectures can be combined into a single network with convolutional layers followed by recurrent layers, often referred to as CRNN. The CRNN makes use of the CNN architecture for the task of feature extraction while using LSTM and GRU placed at the end of the architecture to summarise

the temporal information of the extracted features. The main drawback of CNNs is it lacks longer temporal context information. However, RNNs do not easily capture the invariance in the frequency domain, rendering high-level modeling of the data more difficult [5]. In the C-LSTM and C-GRU architectures, batch normalization is employed after convolutional layers to improve the training speed and performance. Two bidirectional LSTM/GRU units are connected after the time-distributed flatten layer. The bidirectional RNN is preferred rather than unidirectional RNN since it considers the future timestamp representations also [8]. The CNN and CRNN networks are trained using Adam optimizer with a learning rate of 0.001.

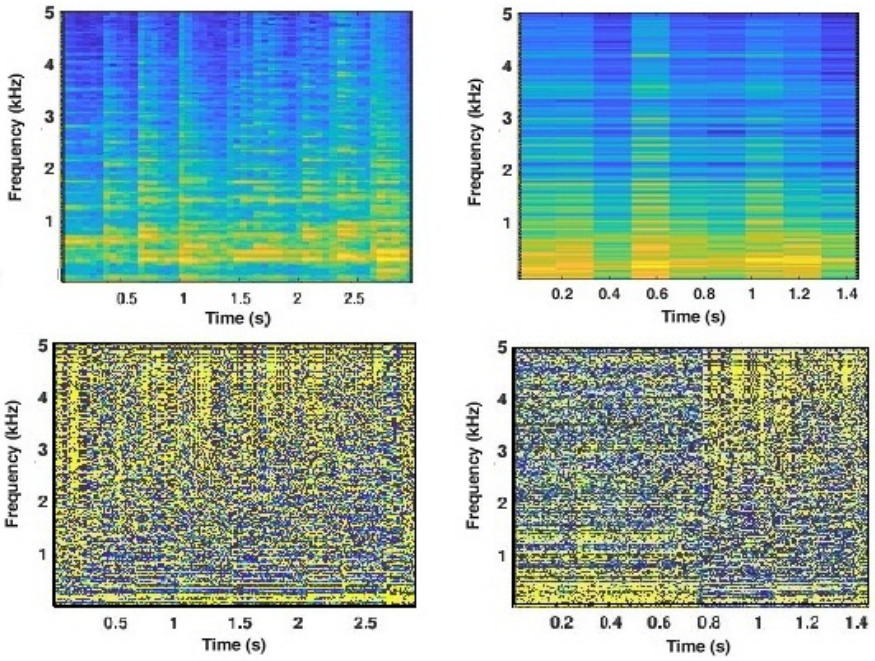


Fig. 2. Visual representation of an audio excerpt with acoustic guitar as leading, Mel-spectrogram of original and WaveGAN-generated (Upper pane left and right). Modgdgram of original and WaveGAN-generated (Lower pane left and right).

A DNN framework on musical texture features (MTF) is also experimented with to examine the performance of deep learning methodology on handcrafted features. MTF includes MFCC-13 dim, spectral centroid, spectral bandwidth, root mean square energy, spectral roll-off, and chroma STFT. The features are computed with a frame size of 40 ms and a hop size of 10 ms using Librosa framework [1]. DNN consists of seven layers, with increasing units from 8 to 512. ReLU has been chosen for hidden layers

[1] https://librosa.org/doc/latest/tutorial.html

and softmax for the output layer. The network is trained using categorical cross-entropy loss function for 500 epochs using Adam optimizer with a learning rate of 0.001.

5 Performance Evaluation

5.1 Dataset

IRMAS dataset [11], comprising eleven classes, is used for the evaluation. The classes include cello (Cel), clarinet (Cla), flute (Flu), acoustic guitar (Gac), electric guitar (Gel), organ (Org), piano (Pia), saxophone (Sax), trumpet (Tru), violin (Vio) and human singing voice (Voice). The training data are single-labeled and consists of 6705 audio files with excerpts of 3 s from more than 2000 distinct recordings. On the other hand, the testing data are multi-labeled and consist of 2874 audio files with lengths between 5 s and 20 s and contain the presence of multiple predominant instruments.

5.2 Data Augmentation using WaveGAN

WaveGAN v2 is used here to generate polyphonic files with the leading instrument required for training. WaveGAN is similar to DCGAN, which is used for Mel-spectrogram generation, in various music processing applications. The transposed convolution operation of DCGAN is modified to widen its receptive field in WaveGAN. For training, the WaveGAN optimizes WGAN-GP using Adam for both generator and discriminator. A constant learning rate of 0.0001 is used with $\beta_1 = 0.5$ and $\beta_2 = 0.9$ [10]. WaveGAN is trained for 2000 epochs on the three sec audio files of each class to generate similar audio files and a total of 6585 audio files with cello (625), clarinet (482), flute (433), acoustic guitar (594), electric guitar (732), organ (657), piano (698), saxophone (597), trumpet (521), violin (526) and voice (720) are generated. The generated files are denoted by $Train_g$ and training files available in the corpus are denoted by $Train_d$. Mel-spectrogram and modgdgram of natural and generated audio files for acoustic guitar are shown in Fig. 2. The experiment details and a few audio files can be accessed at https://sites.google.com/view/audiosamples-2020/home/instrument

The quality of generated files is evaluated using a perception test. It is conducted with ten listeners to assess the quality of generated files for 275 files covering all classes. Listeners are asked to grade the quality by choosing one among the five opinion grades varying from poor to excellent quality (scores, 1 to 5). A mean opinion score of 3.64 is obtained. This value is comparable to the mos score obtained in [10] and [3] using WaveGAN.

5.3 Experimental Set-up

The experiment is progressed in three phases namely Mel-spectrogram-based, modgdgram-based, and score-level fusion-based. Han's sliding window baseline model [16] is implemented for the given experiment with 1 s slice length for performance comparison [2]. We used the same aggregation strategy (S2) as that of Han's model, by summing

[2] https://github.com/Veleslavia/EUSIPCO2017

Table 2. F1 score for the experiments with data augmentation ($Train_d + Train_g$).

SL. No	Class	MTF DNN	Han's Model	Fusion CNN	Fusion LSTM	Fusion GRU	Fusion C-LSTM	Fusion C-GRU
		F1	F1	F1	F1	F1	F1	F1
1	Cel	0.15	0.55	0.55	0.15	0.36	0.42	0.50
2	Cla	0.26	0.18	0.36	0.13	0.36	0.48	0.39
3	Flu	0.27	0.43	0.55	0.32	0.62	0.34	0.31
4	Gac	0.43	0.72	0.63	0.44	0.54	0.51	0.70
5	Gel	0.36	0.69	0.67	0.50	0.49	0.62	0.74
6	Org	0.28	0.45	0.55	0.37	0.49	0.66	0.51
7	Pia	0.36	0.67	0.62	0.50	0.57	0.78	0.78
8	Sax	0.28	0.61	0.58	0.25	0.55	0.47	0.50
9	Tru	0.18	0.44	0.65	0.33	0.62	0.43	0.60
10	Vio	0.22	0.48	0.68	0.38	0.49	0.64	0.69
11	Voice	0.32	0.85	0.73	0.60	0.58	0.85	0.88
	Macro	**0.28**	**0.55**	**0.60**	**0.36**	**0.52**	**0.56**	**0.60**
	Micro	**0.32**	**0.64**	**0.65**	**0.43**	**0.55**	**0.65**	**0.69**

all the softmax predictions followed by normalization and applying a threshold of 0.5. Mel-spectrograms and modgdgrams of input size 128x100x1, corresponding to a window size of 1 s are applied to the corresponding network. The experiments are repeated for CNN, RNN with LSTM and GRU, CRNN with C-LSTM, and C-GRU respectively. Since the number of annotations for each class was not equal, we computed precision, recall, and F1 measures for both the micro and the macro averages. For the micro averages, we calculated the metrics globally, thus giving more weight to the instrument with a higher number of appearances. On the other hand, we calculated the metrics for each label and found their unweighted average for the macro averages.

6 Results and Analysis

Several studies [30, 29] have demonstrated that by consolidating information from multiple sources, better performance can be achieved than uni-modal systems which motivated us to perform the score-level fusion. The standard metrics for various algorithms on the IRMAS corpus are reported in Table 3. Fusion network C-GRU achieved micro and macro F1 measures of 0.69 and 0.60, respectively, which is 7.81% and 9.09% higher than those obtained for the state-of-the-art Han's model. Han employed Mel-spectrogram-CNN for the proposed task. Conventionally, the spectrum-related features used in instrument recognition take into account merely the magnitude information. However, there is often additional information concealed in the phase, which could be beneficial for recognition [9]. The experimental results validate the claim in [9]. Our Fusion-CNN with data augmentation reports a micro and macro F1 score of 0.65 and 0.60 respectively which is 1.56% and 5.26% higher than that obtained for our Mel-

Table 3. Performance comparison on IRMAS dataset

SL.No	**Model**	**F1 Micro**	**F1 Macro**
1	Bosch *et al.* [4]	0.50	0.43
2	Han *et al.* [16]	**0.65**	0.50
3	Pons *et al.* [22]	0.65	0.52
4	Kratimenos *et al.* [17]	0.65	**0.55**
5	**MTF-DNN** ($Train_d + Train_g$)	0.32	0.28
6	**Han Model** ($Train_d + Train_g$)	0.64	0.55
7	**Proposed Mel-spectrogram-CNN** ($Train_d + Train_g$)	0.64	**0.57**
8	**Proposed Modgdgram-CNN** ($Train_d + Train_g$)	0.54	0.53
9	**Proposed Fusion-CNN** ($Train_d + Train_g$)	**0.65**	**0.60**
10	**Proposed Fusion-C-LSTM** ($Train_d + Train_g$)	**0.65**	**0.56**
11	**Proposed Fusion-C-GRU** ($Train_d$)	0.62	0.53
12	**Proposed Mel-spectrogram-C-GRU** ($Train_d + Train_g$)	**0.66**	**0.59**
13	**Proposed Modgdgram-CGRU** ($Train_d + Train_g$)	0.55	0.53
14	**Proposed Fusion-C-GRU** ($Train_d + Train_g$)	**0.69**	**0.60**

spectrogram-CNN with data augmentation. It is evident that modgdgram added complementary information to the spectrogram approach and the importance of the fusion framework for the proposed task. Han's model and the proposed Mel-spectrogram-CNN approach show similar performance with better performance for the proposed architectural choice.

The F1 score of different fusion experiments is tabulated in Table 2. Fusion experiments using RNNs alone do not show improved performance over existing algorithms, however, GRU shows better performance than LSTM. Since we employed the same number of hidden units for both, GRU required less number of trainable parameters and makes faster progress, and reaches the convergence earlier than LSTM. Fusion experiments C-LSTM and CNN show similar performance, but C-GRU outperforms all the models. GRUs train faster and computationally more efficient than LSTM because of fewer trainable parameters. Results of the experiments described in [7] suggest that GRUs perform better than LSTMs on small polyphonic dataset [7]. Our C-LSTM for Mel-spectrogram requires 100224 more trainable parameters compared to C-GRU. It reaches convergence faster without compromising accuracy. The experimental results validate the claim in [7].

Our best model Fusion C-GRU, without data augmentation ($Train_d$) reports micro and macro F1 score of 0.62 and 0.53 respectively. Fusion C-GRU ($Train_d + Train_g$) reports micro and macro F1 scores of 0.69 and 0.60, respectively, with an improvement of 11.29% and 13.21% higher than that obtained by Fusion C-GRU ($Train_d$). This shows the significance of data augmentation in the proposed task.

Our proposed CRNN technique outperformed existing algorithms on the IRMAS dataset for both the micro and the macro F1 measures. The analysis of the experimental frameworks shows the significance of CRNN architecture for the proposed task. Be-

sides, the experiments show the potential of fusion of magnitude and phase information in the proposed task.

7 Conclusion

We presented a CRNN-based predominant instrument recognition system using Mel-spectro/modgd-gram. CRNN is used to capture the instrument-specific characteristics and then do further classification. The proposed method is evaluated on IRMAS dataset. Data augmentation is also performed using WaveGAN. The results show the potential of C-GRU architecture on the score-level fusion of Mel-spectrogram and modgdgram in the proposed task.

References

1. Ajayakumar, R., Rajan, R.: Predominant instrument recognition from polyphonic music using feature fusion. in Proc. of Emerging Trends in Engineering, Science and Technology for Society, Energy and Environment pp. 745–750 (2018)
2. Ajayakumar, R., Rajan, R.: Predominant instrument recognition in polyphonic music using gmm-dnn framework. in Proc.of International Conference on Signal Processing and Communications (SPCOM) pp. 1–5 (2020)
3. Atkar, G., Jayaraju, P.: Speech synthesis using generative adversarial network for improving readability of hindi words to recuperate from dyslexia. Neural Computing and Applications pp. 1–10 (2021)
4. Bosch, J.J., Janer, J., Fuhrmann, F., Herrera, P.: A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. in Proc. of 13th International Society for Music Information Retrieval Conference (ISMIR) (2012)
5. Cakır, E., Parascandolo, G., Heittola, T., Huttunen, H., Virtanen, T.: Convolutional recurrent neural networks for polyphonic sound event detection. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**(6), 1291–1303 (2017)
6. Choi, K., Fazekas, G., Sandler, M., Cho, K.: Convolutional recurrent neural networks for music classification. in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) pp. 2392–2396 (2017)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. NIPS 2014 Workshop on Deep Learning, December 2014 (2014)
8. Cui, Z., Ke, R., Pu, Z., Wang, Y.: Deep bidirectional and unidirectional lstm recurrent neural network for network-wide traffic speed prediction. arXiv preprint arXiv:1801.02143 (2018)
9. Diment, A., Rajan, P., Heittola, T., Virtanen, T.: Modified group delay feature for musical instrument recognition. in Proc. of 10th International Symposium on Computer Music Multidisciplinary Reserach, Marseille, France pp. 431–438 (May 2013)
10. Donahue, C., McAuley, J., Puckette, M.: Adversarial audio synthesis. in Proc. of International Conference on Learning Representations pp. 1–16 (2019)
11. Fuhrmann, F., Herrera, P.: Polyphonic instrument recognition for exploring semantic similarities in music. in Proc. of 13th International Conference on Digital Audio Effects DAFx10, Graz, Austria **14**(1), 1–8 (2010)
12. Fuhrmann, F., et al.: Automatic musical instrument recognition from polyphonic music audio signals. Ph.D. thesis, Universitat Pompeu Fabra (2012)

13. Gimeno, P., Viñals, I., Ortega, A., Miguel, A., Lleida, E.: Multiclass audio segmentation based on recurrent neural networks for broadcast domain data. EURASIP Journal on Audio, Speech, and Music Processing **2020**(1), 1–19 (2020)
14. Gómez, J.S., Abeßer, J., Cano, E.: Jazz solo instrument classification with convolutional neural networks, source separation, and transfer learning. in Proc. of International Society for Music Information Retrieval (ISMIR) pp. 577–584 (2018)
15. Gururani, S., Summers, C., Lerch, A.: Instrument activity detection in polyphonic music using deep neural networks. in Proc. of International Society for Music Information Retrieval Conference (ISMIR) pp. 577–584 (2018)
16. Han, Y., Kim, J., Lee, K.: Deep convolutional neural networks for predominant instrument recognition in polyphonic music. IEEE/ACM Transactions on Audio, Speech, and Language Processing **25**(1), 208–221 (2017)
17. Kratimenos, A., Avramidis, K., Garoufis, C., Zlatintsi, A., Maragos, P.: Augmentation methods on monophonic audio for instrument classification in polyphonic music. in Proc. of 28th European Signal Processing Conference (EUSIPCO) pp. 156–160 (2021)
18. Li, X., Wang, K., Soraghan, J., Ren, J.: Fusion of hilbert-huang transform and deep convolutional neural network for predominant musical instruments recognition. in Proc. of 9th International conference on Artificial Intelligence in Music, Sound, Art and Design (2020)
19. Murthy, H.A., Yegnanarayana, B.: Group delay functions and its application to speech processing. Sadhana **36**(5), 745–782 (2011)
20. Nasrullah, Z., Zhao, Y.: Music artist classification with convolutional recurrent neural networks. in Proc.of International Joint Conference on Neural Networks (IJCNN) pp. 1–8 (2019)
21. O'shaughnessy, D.: Speech communication: human and machine. Universities press pp. 1–5 (1987)
22. Pons, J., Slizovskaia, O., Gong, R., Gómez, E., Serra, X.: Timbre analysis of music audio signals with convolutional neural networks. in Proc. of 25th European Signal Processing Conference (EUSIPCO) pp. 2744–2748 (2017)
23. Rajan, R., Murthy, H.A.: Two-pitch tracking in co-channel speech using modified group delay functions. Speech Communication **89**, 37–46 (2017)
24. Rajan, R., Murthy, H.A.: Group delay based melody monopitch extraction from music. in Proc.of IEEE International Conference on Acoustics, Speech and Signal Processing (ICAASP) pp. 186–190 (2013)
25. Rajan, R., Murthy, H.A.: Melodic pitch extraction from music signals using modified group delay functions. in Proc. National Conference on of the Communications (NCC) pp. 1–5 (February 2013)
26. Rajan, R., Murthy, H.A.: Music genre classification by fusion of modified group delay and melodic features. in Proc.of Twenty-third National Conference on Communications (NCC) pp. 1–6 (2017)
27. Rajesh, S., Nalini, N.: Musical instrument emotion recognition using deep recurrent neural network. Procedia Computer Science **167**, 16–25 (2020)
28. Reghunath, L.C., Rajan, R.: Attention-based predominant instruments recognition in polyphonic music. in Proc. of 18th Sound and Music Computing Conference (SMC) pp. 199–206 (2021)
29. Toh, K., Jiang, X., Yau, W.: Exploiting global and local decisions for multimodal biometrics verification. IEEE Transactions on Signal Processing pp. 3059–3072 (2004)
30. Wang, Y., Tan, T., Jain, A.: Combining face and iris biometrics for identity verification. in Proc. of Fourth International Conference on AVBPA, Guildford, U.K pp. 805–813 (2003)
31. Yu, D., Duan, H., Fang, J., Zeng, B.: Predominant instrument recognition based on deep neural network with auxiliary classification. IEEE/ACM Transactions on Audio, Speech, and Language Processing **28**, 852–861 (2020)

A Polytemporal Model for Musical Scheduling

Martin Fouilleul, Jean Bresson, and Jean-Louis Giavitto

STMS – Sorbonne Université, IRCAM, CNRS
martin.fouilleul@ircam.fr
jean.bresson@ircam.fr
jean-louis.giavitto@ircam.fr

Abstract. This paper describes the temporal model of a scheduler geared towards show control and live music applications. This model relies on multiple inter-related temporal axes, called timescales. Timescales allow scheduling computations using abstract dates and delays, much like a score uses symbolic positions and durations (e.g. bars, beats, and note values) to describe musical time. Abstract time is ultimately mapped onto wall-clock time through the use of time transformations, specified as tempo curves, for which we provide a formalism in terms of differential equations on symbolic position. In particular, our model allows specifying tempo both as a function of time or as a function of symbolic position, and allows piecewise tempo curves to be built from parametric curves.

Keywords: Symbolic time, Time transformations, Tempo curves, Scheduling

1 Introduction

Timing is of utmost importance in performing arts. Among them, music has developed particularly fine-grained temporal constructs, using both continuous and discrete abstract representations of time. As such, it presents specific and interesting challenges with regard to the composition and interpretation of time at multiple scales, and across multiple independent time-flows.

In this paper we present the temporal model of *Jiffy*, a polytemporal scheduler which is part of an ongoing effort to build a programmable show-controller system for performing arts and interactive multimedia installations[1]. In particular, our temporal model allows specifying tempo either as a function of time or as a function of symbolic position, and allows piecewise tempo curves to be built from parametric curves such as Béziers curves, which are both versatile and intuitive. The scheduler exposes an interface based on fibers, that makes it easy to organize inter-dependant streams of related events.

We first highlight the importance of symbolic time in musical applications (section 2). We then cover the notion of time transformations, and give a differential equation formulation to tempo curves (section 3). We then show how tempo curves equations are solved in Jiffy (section 4). Finally, we present the interface of the scheduler (section 5).

[1] The source code of the scheduler as of the time of writing can be found at https://github.com/martinfouilleul/jiffy_scheduler_standalone/tree/fff78cd5ca6ab895ba3107439e8ac9541811590a.

2 Symbolic Time in Musical Applications

2.1 Common Paradigms in Show-Control Applications

Show controllers are programs used by sound and lighting engineers to create and run temporal scenarios synchronized to the actions of performers on stage. They allow users to launch sound and video samples, control mixing and lighting desks, operate motors for mechatronic stage props, and so on. Several approaches can be identified as to how they present and organize the temporal relations between the cues of the show:

- Timelines, which organize cues on a common, static time axis. Most sequencers, such as ProTools[2] or Cubase[3] fall in that category.
- Cuelists, which organize cues in nested lists with associated timing semantics. Notable examples are QLab[4] or Linux Show Player[5].
- Hybrid models offer both cuelists and timelines, either through separate modes of operation, as in Medialon[6] or Smode[7], or as dual views of the same cues, as in Ableton Live[8].
- Graphical planning environments that allow users to position cues in some abstract space, which maps to time through the use of trajectories, as in Iannix [6], or flow graphs, as in Ossia Score [5].

Despite the diversity of approaches, most show controllers lack an abstract notion of musical time, as they directly map cues to wall-clock dates or to external triggers. Furthermore, musical time is often deployed throughout a work at different scales (e.g. movements, phrases, cells, notes…), and not every scale is tied to the same global tempo, e.g. ornaments such as grace notes and *appogiatura* are not affected in the same way by a change of tempo as a main melody line. Hence it would be more appropriate to consider several abstract musical times, or *timescales*.

2.2 Abstract Timescales

The above discussion emphasizes the need of strong temporal models in composition and performance softwares and highlights the adequacy of *polytemporal abstract time scheduling*, i.e. the ability to organize concurrent computations along multiple logical timescales, that can later get mapped to wall-clock time.

[2] https://www.avid.com/pro-tools
[3] https://new.steinberg.net/cubase/
[4] https://qlab.app/docs/QLab_4_Reference_Manual.pdf
[5] https://linux-show-player-users.readthedocs.io/en/latest/index.html
[6] https://medialon.com/wp-content/uploads/2019/07/M515-1-Medialon-Control-System-Manual.pdf
[7] https://smode.fr
[8] https://www.ableton.com/en/live/what-is-live/

The notion of abstract timescales has been tackled before by computer music environments or score followers. For instance, FORMULA [1] allows applying independent time deformations on groups of concurrent tasks. David A. Jaffe [9] proposed a recursive scheduler for hierarchical timing control, using explicit time maps. Antescofo [7] allows users to compose independent abstract times through the use of *time scopes* and tempo curves.

In Jiffy, a timescale is a data structure used to maintain a notion of logical time, expressed as a rational number of *symbolic time units*[9] (STU), and to schedule events at specific logical dates. It is analogous in this respect to a score, which organizes musical events in terms of a musical time, that needs to be translated into wall-clock time by a musician according to tempo indications and interpretative choices.

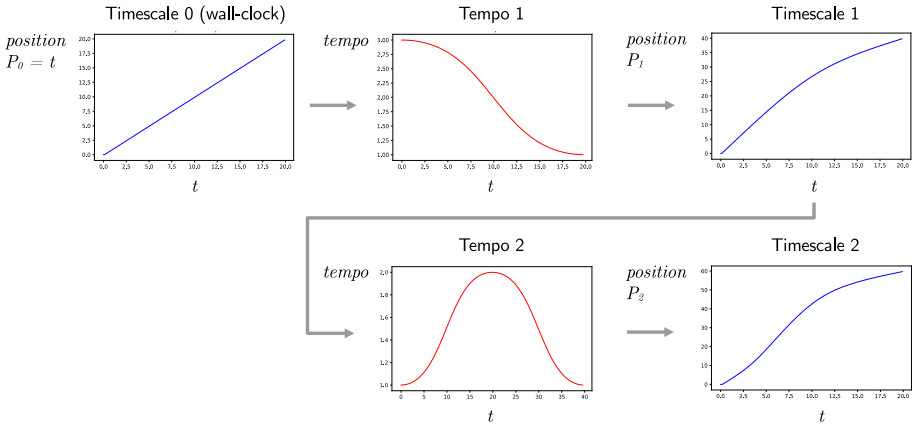


Fig. 1. Composing time deformations using tempo curves.

However, whereas the tempo indication of a score usually prescribes some idealized mapping from musical-time to wall-clock time, a timescale's logical time does not necessarily map directly to wall-clock time. Instead, each timescale has a *time source*, which can be either the wall-clock time or another timescale. A timescale is also associated with a *time transformation*, which maps its internal time to the time of its source. Thus, the scheduler can handle multiple notions of logical time and map dates to wall-clock time through a hierarchy of time transformations.

Figure 1 illustrates a time deformation between a timescale and its source. The time map plots for each timescale show the position of the timescale with respect to wall-clock time. The effect of the first tempo curve (tempo 1) is to warp the time map of

[9] We deliberately avoid the term *beats* here. We think it would bring some confusion by conflating the notion of time unit with the notion of meter, and by suggesting that all beats are of equal conceptual length. This is, in fact, rather a Western exception than a universal norm.

timescale 0 (which represents wall-clock time) into that of timescale 1 (which represent some abstract musical time). Timescale 1 is then transformed by another another tempo curve (tempo 2), to produces the time map of timescale 2.

3 Time Transformations

Jiffy's scheduler must be able to transform timescale-local positions to and from wall-clock time. These transformations are specified by the means of tempo curves, which describe the speed of a timescale's "playhead" with respect to the source time, much like tempo indications in a score prescribe an idealized conversion from durations in beats to durations in wall-clock time[10].

Several methods have been proposed to represent time transformations and to integrate tempo curves to map symbolic position to time. Jaffe [9] proposes to directly use time maps constructed from a collection of predefined time warping functions. Berndt [2] chooses to represent tempo curves by potential functions of symbolic position, matching some specified mean tempo condition. Timewarp [10] is a tool that uses regularized beta functions to define tempo curves satisfying polyrhythmic constraints. Antescofo uses a variety of tweening functions[11] to express tempo as a function of time, and uses closed form expressions to compute time transformation based on tempo curves. When there is no analytical solution to a tempo curve integration, Antescofo samples the curve to produce a piecewise linear approximation, which is then integrated analytically. Antescofo can also use arbitrary expressions to define tempo, although these expressions are not integrated: they are reevaluated each time a variable is updated, and considered constant between updates. As such, they can only represent tempo as step functions.

In Jiffy, we allow users to specify a tempo curve either as a function of a timescale's source time, or as a function of symbolic position (which is closer to the way tempo is specified in a score). We use piecewise tempo curves where each piece can be defined by parametric curves. A variable-step numerical method is used to integrate the tempo curves when simple analytical solutions are not readily available.

3.1 Differential Equation Formulation

In the following we will use the variable p to denote the position in a timescale, i.e. the logical time in this timescale's reference frame. The variable t will be used to denote the source time (or simply, *time*), i.e. the time in the timescale's parent reference frame (which could be the wall-clock time).

The function *position function*, $P(t)$, transforms the source time into the internal position of the timescale. The *time function*, $T(p)$, transforms the position into the source time. Obviously, $P = T^{-1}$.

[10] One difference, however, is that we use the word tempo here to refer to the ratio of internal STUs over source STUs, rather than the number of beats per minutes, since the latter could depend on the musical meter of the timescale.

[11] `https://antescofo-doc.ircam.fr/Reference/compound_curve/`

A *tempo curve* $\boldsymbol{\mathcal{T}}$ can be either a function of time or position. It maps its parameter to the value of the derivative of the position function at this instant. In the following we will refer to a tempo curve defined as a function of position as an *autonomous tempo curve*, whereas a tempo curve defined as a function of time will be referred to as a *non-autonomous tempo curve*. This naming stems from the formulation of the tempo curve as the right-hand side of an autonomous or non-autonomous differential equation:

$$\frac{dP}{dt}(t) = \boldsymbol{\mathcal{T}}(P(t)) \quad \text{(autonomous)}\,, \text{ or } \frac{dP}{dt}(t) = \boldsymbol{\mathcal{T}}(t) \quad \text{(non autonomous)}\,, \\ \text{with initial condition } P(0) = 0\,. \tag{1}$$

4 Tempo Curves Integration

Tempo curves in Jiffy are defined as piecewise functions. For the sake of brevity, we may refer to an interval and its associated sub-function as a tempo curve *segment*, or simply as a *curve*, where the meaning should be clear from context. Each segment is defined by a start tempo and an end tempo, a duration, an interpolation mode and optional interpolation parameters. We implemented three interpolation modes, namely *constant*, *linear* and *parametric*.

4.1 Integration of Constant and Linear Tempo Curves

Constant and linear tempo curves can be solved analytically. We show below the differential equation of tempo, and the position and time functions for each case.

Constant Tempo.

$$\boldsymbol{\mathcal{T}}(p) = \boldsymbol{\mathcal{T}}_0\,. \tag{2}$$

$$T(p) = \frac{p}{\boldsymbol{\mathcal{T}}_0}\,,\; P(t) = t \times \boldsymbol{\mathcal{T}}_0\,. \tag{3}$$

Autonomous Linear Tempo.

$$\boldsymbol{\mathcal{T}}(p) = \boldsymbol{\mathcal{T}}_0 + \alpha p\,, \text{ where } \alpha = \frac{\boldsymbol{\mathcal{T}}_1 - \boldsymbol{\mathcal{T}}_0}{L}\,. \tag{4}$$

$$P(t) = \frac{\boldsymbol{\mathcal{T}}_0}{\alpha}(e^{\alpha t} - 1)\,, \text{ and } T(p) = \frac{1}{\alpha}\log(1 + \frac{\alpha p}{\boldsymbol{\mathcal{T}}_0})\,. \tag{5}$$

Non-autonomous Linear Tempo.

$$\boldsymbol{\mathcal{T}}(t) = \boldsymbol{\mathcal{T}}_0 + \alpha t\,, \text{ where } \alpha = \frac{\boldsymbol{\mathcal{T}}_1 - \boldsymbol{\mathcal{T}}_0}{L}\,. \tag{6}$$

$$P(t) = \boldsymbol{\mathcal{T}}_0 t + \frac{\alpha}{2}t^2\,, \text{ and } T(p) = \frac{\sqrt{\boldsymbol{\mathcal{T}}_0^2 + 2\alpha p} - \boldsymbol{\mathcal{T}}_0}{\alpha}\,. \tag{7}$$

Numerical Considerations Some of the above time and position functions are indeterminate forms for $\alpha \to 0$. To avoid that problem, we approximate these expressions by a series expansions in α when $|\alpha|$ is smaller than a given threshold. For instance, our approximation of the position function for the autonomous case when is $|\alpha| < 10^{-9}$ is:

$$P(t) \approx \boldsymbol{\mathcal{T}}_0(t + \frac{\alpha}{2}t^2 + \frac{\alpha^2}{6}t^3 + \frac{\alpha^3}{24}t^4 + \frac{\alpha^4}{120}t^5)\,. \tag{8}$$

4.2 Parametric tempo curves.

In this section we will give a definition of a parametric tempo curve, and show the differential equations that need to be solved in order to compute the time and position functions. These equations are then solved by a numerical solver.

An autonomous (resp. non-autonomous) parametric tempo curve segment is defined as a function $\mathcal{C}$ of the position p (resp. of the time t), which describes the same curve in the plane $(p, \boldsymbol{\mathcal{T}})$ (resp. $(t, \boldsymbol{\mathcal{T}})$) as a parametric curve $\boldsymbol{B}(s)$ with components $B_x(s)$ and $B_y(s)$.

Autonomous Parametric Tempo. The differential equation corresponding to an autonomous tempo curve can be written as

$$\frac{dP}{dt}(t) = C(P(t))\,. \tag{9}$$

Position function $P(t)$. The derivative of the position with respect to time is directly expressed by the autonomous tempo curve,

$$\frac{dP}{dt}(t) = B_y(s)\,,\ \text{where } s = B_x^{-1}(P(t))\,. \tag{10}$$

Time function $T(p)$. We operate the change of variable $s = B_x^{-1}(p)$ on Equation 9. Finding the time function is then a matter of solving the differential equation

$$\frac{d\tilde{T}}{ds}(s) = \frac{B_x'(s)}{B_y(s)}\,,\ \text{with } \tilde{T}(s) = T(p)\,. \tag{11}$$

Non-autonomous Parametric Tempo. The definition of the non-autonomous parametric tempo curves can be written as

$$\frac{dP}{dt}(t) = C(t)\,. \tag{12}$$

Position function $P(t)$. Using the change of variable $s = B_x^{-1}(t)$ and the chain rule, we can write the differential equation for the position function as

$$\frac{d\tilde{P}}{ds}(s) = B_y(s)B_x'(s)\,,\ \text{with } \tilde{P}(s) = P(t)\,. \tag{13}$$

Time function $T(p)$. Using the formula for the derivative of inverse functions on Equation 12, we get

$$\frac{dT}{dp}(p) = \frac{1}{\mathcal{C}(T(p))} = \frac{1}{B_y(s)}\,, \text{ where } s = B_x^{-1}(T(p))\,. \tag{14}$$

Numerical Resolution. Although some of the above equations can be solved analytically, using a numerical solver has the advantage of allowing us to control the tradeoff between accuracy and speed, and opens up the possibility of supporting other arbitrary functions to define tempo curves. We use a Cash-Karp [4] solver to numerically solve the tempo curve equations. We follow the general architecture proposed in [11], optimized further by leveraging the fact that these equations are either autonomous or directly integrable.

Bézier Tempo Curves. The above formulation allows the use of any parametric curve, provided that it describes a derivable, non null function. Our specific implementation uses cubic Bézier curves, which are especially versatile, as they allow putting constraints on both endpoints and their first derivative, while ensuring that the curve remains contained inside its control points' convex hull. They are also intuitive to manipulate and map well to the curve-editing interfaces commonly used in animation, audio, and video applications.

An autonomous (resp. non-autonomous) Bézier tempo curve segment is defined by the parametric curve

$$\boldsymbol{B}(s) = \boldsymbol{C}_3 s^3 + \boldsymbol{C}_2 s^2 + \boldsymbol{C}_1 s + \boldsymbol{C}_0\,, \tag{15}$$

where the $\boldsymbol{C}_i$ are the power basis coefficients computed from the Bézier curve's control points. To ensure that the curve describes a function, the cubic function $\boldsymbol{B}_x(s)$ must be monotonous, i.e. if the x_i are the abscissae of the $\boldsymbol{C}_i$, the condition $c_1^2 - 3c_0c_2 \leq 0$ must hold.

Bézier curves evaluation. We should stress out that, although each coordinate of the parametric Bézier curve is cubic with respect to its parameter s, the second coordinate is *not* a cubic function of the first, i.e. the tempo is not a cubic function of position (resp. time). Analytically finding the tempo for a given position (resp. time) indeed requires solving a third order equation.

A faster method is to numerically find the parameter s for a given position (resp. time), up to some desired precision, and then compute the tempo from s. Our implementation first uses the Newton-Raphson root-finding method up to a fixed number of iterations, and falls back to a bisection algorithm if either the value of the derivative falls behind some threshold, or the desired precision is not reached within the maximum iteration count.

An example of a time map produced by tempo curve composed of two Bézier segments is shown in Figure 2. The blue curve shows position as a function of time. The orange stems mark the timeline STUs. The red curve shows the tempo curve, as a function of

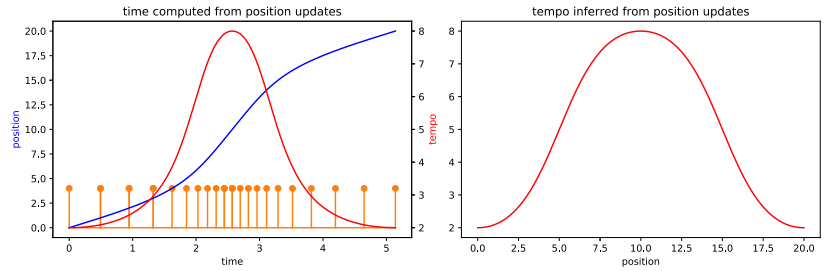


Fig. 2. Time map and beats trace for a tempo curve defined by two Bézier curves.

time (on the left), or as a function of position (on the right). The figure is produced by computing the positions corresponding to a regularly spaced time grid.

5 Scheduler Interface

The Jiffy scheduler is designed to run user code in fibers[12]. Compared to callback-based scheduling APIs (such as [3], [13] or [12]), this doesn't compel the user to break down the control flow of their code into lots of small functions, keeps logically related computations in the same local context, and allows users to easily express dependencies between several workloads. Fibers can also be migrated between threads, allowing a very streamlined way to handle blocking calls without hanging the scheduler.

The scheduler uses the notion of *tasks* to represent a group of fibers executing within the same timescale. Tasks, like timescales, are organized in a parent-children relationship. The API exposes functions to launch new tasks and fibers, to yield and reschedule the current fiber to a future date, or to wait on the completion of other tasks or fibers. It also features functions to move fibers into background jobs to perform blocking operations without blocking the scheduler, and bringing them back to the foreground once done.

Listing 1.1 shows a simple example that launches a task to print a message at regular symbolic intervals with a varying tempo. This task lives for 40 time units unless it is canceled from another fiber, which waits for user input in the background.

6 Conclusion and Future Work

In this paper, we highlighted the need for symbolic time scheduling in show-control software and musical applications. We then described a temporal model based on time

[12] The notions of *fiber*, *coroutine*, or *green thread* are so closely related that the distinction between them, if any, is amenable to debate. One could argue that *green thread* is more appropriate in the context of a virtual machine or runtime environment, while *coroutine* originates from programming language design. The term *fiber* may capture a more general view of the concept.

```
i64 my_task_proc(void* userPointer)
{
        // main task: print a message at each symbolic time unit
        for(int i=0; i<40; i++)
        {
                printf("Hello, world: %i\n", *count);
                sched_wait(1);
        }
        return(0);
}

i64 user_cancel_fiber(void* userPointer)
{
        // go to background so we don't block the scheduler, and wait user input,
        // then bring the fiber to the foreground and cancel the main task
        sched_background();
        while(getchar() != 'q') /* wait 'quit' command */ ;
        sched_foreground();
        sched_task_cancel(*(sched_task*)userPointer);
        return(0);
}

int main()
{
        // launch our main task and apply a tempo curve to it, then launch the
        // user canceling fiber, and wait for the main task to complete.

        sched_curve_descriptor_elt elements[2] = {
                {.type = SCHED_CURVE_BEZIER,
                 .startValue = 2, .endValue = 8, .length = 20,
                 .p1x = 0.5, .p1y = 0, .p2x = 0.5, .p2y = 1},
                {.type = SCHED_CURVE_BEZIER,
                 .startValue = 8, .endValue = 2, .length = 20,
                 .p1x = 0.5, .p1y = 0, .p2x = 0.5, .p2y = 1}};

        sched_curve_descriptor desc = {.axes = SCHED_CURVE_POS_TEMPO,
                                       .eltCount = 2, .elements = elements};
        sched_init();
        sched_task task = sched_task_create(my_task_proc, 0);
        sched_task_timescale_set_tempo_curve(task, &desc);

        sched_create_fiber(user_cancel_fiber, &task, 0);

        sched_wait_completion(task);
        sched_end();
        return(0);
}
```

Listing 1.1. An example of Jiffy's scheduling API.

transformations expressed through tempo curves, and gave a formalism of such curves. We then described how these curves are implemented in the Jiffy scheduler, and presented the API of the scheduler.

In its current form, the scheduler is a local system, only maintaining proper time flow for its host process. Synchronizing timescales across multiple scheduler instances (potentially running on different machines) is the subject of ongoing work.

The kind of synchronization we considered in this paper was only concerned about relative speeds. However, when dealing with ensemble music, the notion of synchronization is really about the relative phase of each musician. We could refer to this type of synchronization as *metric synchronization*. Ableton Link [8] is one of the tools that tackle this problem, and offers an elegant model to build musical structure on top of beat synchronization. However it has some limitations when it comes to multiple tempos and complex polyrhythms. Addressing these scenarios will be the subject of further research.

References

[1] David P. Anderson and Ron Kuivila. "A System for Computer Music Performance". en. In: *ACM Transactions on Computer Systems* 8.1 (1990), pp. 56–82.

[2] A. Berndt. "Musical Tempo Curves". In: *ICMC*. 2011.

[3] Dimitri Bouche and Jean Bresson. "Planning and Scheduling Actions in a Computer-Aided Music Composition System". In: *Scheduling and Planning Applications Workshop (SPARK)*. Ed. by Israel Science Foundation. Proceedings of the 9th International Scheduling and Planning Applications Workshop. Jerusalem, Israel: Steve Chien and Mark Giuliano and Riccardo Rasconi, 2015, pp. 1–6.

[4] J. R. Cash and Alan H. Karp. "A Variable Order Runge-Kutta Method for Initial Value Problems with Rapidly Varying Right-Hand Sides". In: *ACM Trans. Math. Softw.* 16.3 (1990), pp. 201–222.

[5] Jean-Michaël Celerier et al. "OSSIA: Towards a Unified Interface for Scoring Time and Interaction". In: *TENOR 2015 - First International Conference on Technologies for Music Notation and Representation*. Paris, France, 2015.

[6] Thierry Coduys and G. Ferry. "Iannix. Aesthetical/Symbolic Visualisations for Hypermedia Composition". In: *Sound and Music Computing Conference (SMC)*. 2004.

[7] Arshia Cont. "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music." In: *International Computer Music Conference (ICMC)*. Belfast, Ireland, 2008, pp. 33–40.

[8] Florian Goltz. "Ableton Link – A Technology to Synchronize Music Software". en. In: *Proceedings of the Linux Audio Conference (LAC 2018)*. 2018, p. 4.

[9] David Jaffe. "Ensemble Timing in Computer Music". In: *Computer Music Journal* 9.4 (1985), pp. 38–48.

[10] John MacCallum and Andrew Schmeder. "Timewarp: A Graphical Tool for the Control of Polyphonic Smoothly Varying Tempos". en. In: *International Computer Music Conference, ICMC 2010* (2010), p. 4.

[11] William H. Press et al. "Integration of Ordinary Differential Equations". In: *Numerical Recipes in c (2nd Ed.): The Art of Scientific Computing*. USA: Cambridge University Press, 1992, pp. 710–722.

[12] Charles Roberts, Graham Wakefield, and Matthew Wright. "2013: The Web Browser as Synthesizer and Interface". en. In: *A NIME Reader*. Ed. by Alexander Refsum Jensenius and Michael J. Lyons. Vol. 3. Cham: Springer International Publishing, 2017, pp. 433–450.

[13] Norbert Schnell et al. "Of Time Engines and Masters an API for Scheduling and Synchronizing the Generation and Playback of Event Sequences and Media Streams for the Web Audio API". In: *WAC*. Paris, France, 2015.

A Framework for Music Similarity and Cover Song Identification

Roberto Piassi Passos Bodo*[1], Emmanouil Benetos[2], and Marcelo Queiroz[1]

[1] Institute of Mathematics and Statistics, University of São Paulo, Brazil
{rppbodo,mqz}@ime.usp.br
[2] Centre for Digital Music, Queen Mary University of London, UK
emmanouil.benetos@qmul.ac.uk

Abstract. This paper presents a framework for music information retrieval tasks which relate to music similarity. The framework is based on a pipeline consisting of audio feature extraction, feature aggregation and distance measurements, which generalizes previous work and includes hundreds of similarity models not previously considered in the literature. This general pipeline is subjected to a comprehensive benchmark of analogously defined music similarity models over the task of cover song identification. Experimental results provide scientific evidence for certain preferred combined choices of features, aggregations and distances, while pointing towards novel combinations of such elements with the potential to improve the performance of music similarity models on specific MIR tasks.

1 Introduction

Using Music Information Retrieval (MIR) techniques to deal with large sets of music files has become an increasingly common practice. Working directly with audio and musical contents has several advantages. MIR methods can provide users the ability to hum in order to retrieve a melody or to clap to fetch a rhythm, and to use an audio file as query in a search for similar tracks. The goal of MIR is to make music content more accessible and in a more intuitive way [1].

Music similarity plays a central role in several MIR tasks. It is often desirable to define and calculate similarity measures for pairs of music recordings, based on audio contents and also (derived or annotated) metadata. The use of music similarity measures on a music dataset provides a solid foundation for navigation, organization, recommendation, and search [2,3].

Since there is no universally agreed-upon formalized concept of general musical similarity, a fair solution is to look for similarity models which deal with individual aspects of music, such as pitch, rhythm, dynamics and timbre, providing tools for melodic, harmonic, rhythmic, dynamic and timbre-related retrieval tasks, among others. It is important to state explicitly that the notion of music similarity completely depends on the context of the retrieval task at hand, which is usually established by the type of dataset annotations available.

* This study was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and CNPq Grant 307389/2019-7

The literature on audio-based music similarity presents several approaches, including the use of traditional information retrieval methods, such as extracting features from audio recordings and computing their distances within a vector space [4,5,6,7,8,9], modeling the extracted feature distributions and comparing the corresponding statistical models [10,11,12,13,14,15], feature learning [16,17], metric learning [18,19], and deep neural networks [20,21].

The framework presented in this paper, which generalizes the first two approaches above, is based on a conceptual pipeline [3] that breaks down a generic music similarity model into three components (feature extraction, aggregation and distance computing), completely specified by the choices of techniques employed in each component. Its implementation allows the user to freely combine virtually any techniques within each component, thus providing a direct way of experimenting with a large number of similarity models at once.

This possibility is explored in the context of music similarity tasks, including Cover Song Identification (CSI) [22], an application which involves identifying songs[3] which are versions (covers) of each other, assuming that versions of a song should have some common music trait captured by a music similarity model. This paper presents, to the best of the authors' knowledge, the first attempt to comprehensively benchmark music similarity models in music similarity tasks, where hundreds of models not previously considered in the literature are tested.

The main goal of the experiments here presented is to identify which music similarity models lead to best results for the annotated datasets considered, which have been compiled for melodic similarity tasks, rhythmic similarity tasks, genre classification and CSI. Another contribution of this paper is a modular open-source framework[4] for music similarity offering numerous alternatives for feature extraction, aggregation and distances.

The remainder of the paper is organized as follows: Section 2 presents the music similarity models considered; Section 3 presents the metrics considered to assess the discriminating power of the music similarity models; Section 4 presents the experiments, including the selected datasets, the experimental design, the results and their discussion; Section 5 outlines the conclusions and directions for future work.

2 Music Similarity Framework

The music similarity framework considered here implements the following pipeline: 1. extract audio features; 2. aggregate local features into global features; and 3. compute the similarities of every pair of audio recordings within a dataset. A triple $\{extractor_i$, $aggregator_j$, $distance_k\}$ defines a music similarity model, and our main goal is to benchmark music similarity models, identifying which models lead to best results for each annotated dataset. Additionally, the models are also applied to datasets designed for Cover Song Identification (CSI), another similarity-based music retrieval task.

[3] in the CSI literature, *song* is often taken as a synonym of *audio recording*, regardless of containing singing voice or not.

[4] The source code can be found at `https://github.com/rppbodo/music-similarity-framework`.

Papers addressing music similarity related tasks, including CSI, often derive their similarity measurements from tonal features, such as chromagrams [23,24,7,25], tonnetz [26], and symbolic melodic sequences [27,28,29,30,31,32]. Also used in music similarity retrieval tasks are timbre features (e.g., Mel-Frequency Cepstral Coefficients (MFCC) [4,10,11,13,12,15]), spectral features (e.g., spectral centroid, bandwidth, contrast, flatness, etc) [5], rhythmic features (e.g., Rhythm Pattern) [33,6], and amplitude/energy features (e.g., Root Mean Square (RMS)) [34,35].

Among aggregation methods applied in music similarity are simple statistics (such as mean, standard deviation, skewness and kurtosis, see e.g. [15]), computed from the features themselves and their 1st order differences, Gaussian Mixture Models [11,12,13], Vector Quantization [10], Markov Chains [36], Octave and Interval Abstractions [32], and Pitch Contour (using 3-levels [27] and 5-levels [31]).

The computation of the similarity between two audio recordings is based on a chosen distance applied to the (possibly aggregated) features. Distances relevant for music similarity are Manhattan [6,8], Euclidean [4,5,7], Cosine [4,9], Longest Common Subsequence based distances [37,38], Levenshtein [39,38], Kullback-Leibler [13,15], Earth Mover [10,14], and Monte Carlo distances [11,12].

The detailed analysis of the techniques proposed in the music similarity literature allows us to observe that several papers do not explicitly argue as for why a particular extractor (or aggregator, or distance) is selected to solve a particular problem. Even less frequent are arguments about why a specific set of techniques are used in combination (instead of many other plausible alternatives). This prompted us to try to explore hundreds of combinations of extractors, aggregators, and distances that are not considered in the literature. It was thus natural to look at this problem as a benchmark, exhaustively experimenting with a large number of music similarity models.

The current list of music similarity models considered in the implemented framework started out from a large set of features, aggregators and distances appearing in the related literature, which has been modified by including and collecting techniques, but also by discarding techniques by many criteria, including the availability of open-source implementations. The rationale for this specific criterion is to avoid producing implementations that might substantially differ from their original implementations due to ambiguous or insufficiently detailed descriptions. A survey of open-source libraries (such as LibROSA[5], Essentia[6], and RP_extract[7]) led us to include techniques not previously considered in the music similarity literature. The same criteria were applied to aggregator and distance techniques, but in a softer way, since they are usually much simpler to implement.

Due to compatibility issues, not all available features, aggregators and distances can be combined. Framewise numerical features may be aggregated using any statistical aggregation methods, GMM and VQ. Symbolic melodic features use only specific aggregators (octave/interval abstractions, pitch contours and Markov chains). Numerical aggregations (single and multivariate Gaussians, GMM, vector quantization, and Markov chains) can be compared using spatial distances (Euclidean, Manhattan, Chebyshev,

[5] http://librosa.github.io/
[6] http://essentia.upf.edu/
[7] https://github.com/tuwien-musicir/rp_extract

and cosine). Statistical models can be compared using Kullback Leibler, Earth Mover's and Monte Carlo distances, and all symbolic global features can be compared using LCS-based and Levenshtein distances.

All the compatible combinations of features, aggregators, and distances considered result in a total of 690 music similarity models; the complete list is available at `https://rppbodo.github.io/phd/music-similarity-models.html`, along with descriptions of each function.

3 Music Similarity and Cover Song Identification metrics

The most common way to represent a particular music similarity model applied to a particular dataset is the similarity matrix. The i, j position of this matrix contains the similarity between the i-th and the j-th tracks in the dataset. It can be defined from a normalized distance measure as $\text{sim}(t_i, t_j) = 1 - \text{dist}(t_i, t_j)$.

Intra-Inter Class Similarity Ratios (IICSR) When the dataset partitions its recordings into labeled classes (e.g. genres, composers, melodic or rhythmic patterns), we may define the quality of a music similarity model using the intra-inter-class similarity ratio, computed from the similarity matrix according to the following formula:

$$IICSR(c) = \frac{\frac{\sum_{t_1 \in T_c} \sum_{t_2 \in T_c, t_1 \neq t_2} sim(t_1,t_2)}{(|T_c|^2 + |T_c|)}}{\frac{\sum_{t_1 \in T_c} \sum_{t_2 \in T_{C \setminus c}} sim(t_1,t_2)}{(|T_{C \setminus c}||T_c|)}}, \tag{1}$$

where T_c is the set of all recordings in class c and $T_{C \setminus c}$ is the complement of T_c. This measure compares the average similarity within the class c (weighted by the number of recordings in this class) with the average similarity for pairs of recordings in different classes (with one member of the pair in class c). If these ratios are greater than 1, the similarity model may be used to classify pairs of recordings as belonging to the same class or to different classes. Intra-inter-class similarity ratios may be summarized by their weighted average:

$$weighted\_mean\_IICSR = \frac{1}{\sum_{c \in C} |T_c|} \sum_{c \in C} IICSR(c) \times |T_c|, \tag{2}$$

where each class is weighted by its size (number of recordings).

Mean Rank (MR) The Mean Rank is broadly used in the CSI literature [9,40], where queries return ranked lists of cover candidates. MR corresponds to the average position (rank) where the first cover appears in the resulting list.

Mean Reciprocal Rank (MRR) The reciprocal rank (inverse of a rank) [41] converts index positions to the $[0, 1]$ range, where higher values represent covers higher up in the list (topmost ranks). MRR corresponds to the average of the reciprocal ranks, and its inverse may be viewed as the harmonic mean of the original ranks.

dataset	n_{tracks}	$n_{classes}$	annotations
Ballroom	698	10	dance styles names
GTZAN	1000	10	musical genres
IOACAS-QBH	1057	298	ground-truth melody id
Panteli's melody dataset	3000	30	original melody id
Panteli's rhythm dataset	3000	30	original rhythm id
MAST	3104	40	ground-truth rhythm id
1517-Artists	3180	19	musical genres
MIR-QBSH	4479	48	ground-truth melody id
FMA-Small	8000	8	musical genres
Covers80	160	80	original song
YouTubeCovers	350	50	original song
Covers1000	1000	395	original song
Mazurkas	2741	49	mazurka id
SHS9K	9286	143	original song

Table 1. Datasets selected to experiment Music Similarity models.

Median Rank (MDR) The MDR is a robust statistic based on the positions of the first retrieved cover, obtained as the median of the ranks for all queries.

Mean Average Precision (MAP) Kim Falk [42] defines Mean Average Precision in the context of recommender systems, in which users perform queries, and each query returns a list of ranked items. Precision at K (P@k) is the number of relevant items found in the first k items; Average Precision (AP) = $\frac{1}{m}\sum_{k=1}^{m} P@k(u)$, where m is the length of the ranked list, and u is the user performing the query; Mean Average Precision (MAP) = $\frac{1}{|U|}\sum_{u\in U} AP(u)$, where U is the set of all users.

4 Experiments and Results

4.1 Datasets

In this Section we present the datasets used to benchmark models within our music similarity framework. The first part of Table 1 presents datasets designed for various music similarity tasks, and the second part presents the datasets designed specifically for Cover Song Identification.

Three datasets are designed for melodic similarity tasks: MIQ-QBSH[8] and IOACAS-QBH[9] are designed for the query-by-humming task (classes are composed of a reference melody and a set of recordings of people trying to hum it), and Maria Panteli's melody dataset[10] uses synthesis to test similarity models against several melodic transformations.

[8] http://mirlab.org/dataSet/public/MIR-QBSH-corpus.rar
[9] http://mirlab.org/dataSet/public/IOACAS_QBH.rar
[10] https://archive.org/details/panteli_maria_melody_dataset

Three other datasets – Ballroom[11], MAST[12], and Maria Panteli's rhythm dataset[13] — are designed for tasks related to rhythm similarity. The Ballroom dataset is composed of recordings from distinct dance styles; MAST has recordings of students successfully reproducing rhythmic patterns; Maria Panteli's rhythm dataset is composed of different synthesized rhythms subjected to several transformations.

The three remaining datasets in the first part of Table 1 — GTZAN[14], 1517-Artists[15], and FMA-Small[16] -– are annotated with music genres assigned to each recording. Several papers in the literature claim that there is a relationship between genre and timbre [43,44,3], and under this assumption, these datasets could be used to test timbre similarity models.

The second part of Table 1 presents datasets designed for CSI: Covers80[17], YouTube-Covers[18], Covers1000[19], Mazurkas[20], and SHS9K[21]. The latter is a sub-set of the SHS100K[22] dataset crafted by the authors by selecting the original songs that have from 50 to 100 covers.

4.2 Experiment Design

Two experiments are proposed. The goal of the first experiment is to check which music similarity models lead to best results for the selected datasets. In order to accomplish this we run each one of the 9 datasets considered through our music similarity framework, compute the Intra-Inter Class Similarity Ratio (IICSR) for every annotated class within the dataset, and finally compute the weighted mean IICSR for each one of the 690 considered models.

The second experiment has a similar goal to the previous one – to check which music similarity models lead to the best results – but now with CSI datasets considering the specific metrics used in this task. We compute similarity matrices using all 690 models for the 5 CSI datasets, and then calculate the Mean Rank (MR), Mean Reciprocal Rank (MRR), Median Rank (MDR), and Mean Average Precision (MAP).

4.3 Results

The results of the first experiment are organized as follows: the best weighted mean IICSR values for each dataset are presented in Table 2, and the entire list of IICSR values computed in this experiment is published in `https://rppbodo.github.io/phd/experiment_1.html`.

[11] `http://mtg.upf.edu/ismir2004/contest/tempoContest/node5.html`
[12] `https://zenodo.org/record/2620357`
[13] `https://archive.org/details/panteli_maria_rhythm_dataset`
[14] `http://marsyas.info/downloads/datasets.html`
[15] `http://www.seyerlehner.info/index.php?p=1_3_Download`
[16] `https://github.com/mdeff/fma/`
[17] `https://labrosa.ee.columbia.edu/projects/coversongs/covers80/`
[18] `https://sites.google.com/site/ismir2015shapelets/data`
[19] `http://www.covers1000.net/`
[20] `http://www.mazurka.org.uk/`
[21] `https://rppbodo.github.io/phd/shs9k.html`
[22] `https://github.com/NovaFrost/SHS100K`

dataset	mean IICSR	extractor	aggregator	distance
Ballroom	1.33824	spectral_bandwidth	vector_quant.	cosine
GTZAN	1.74945	spectral_contrast	vector_quant.	cosine
IOACAS-QBH	1.13599	pitch_cont._seg.	octave_abst.	lcs_circular_min
Panteli's melody	3.12167	pitch_cont._seg.	interval_abst.	levenshtein_max
Panteli's rhythm	2.82936	chroma_cens	vector_quant.	manhattan
MAST	1.47572	mfcc	vector_quant.	cosine
1517-Artists	1.21584	mfcc	vector_quant.	cosine
MIR-QBSH	1.29421	pitch_cont._seg.	octave_abst.	levenshtein_circular_max
FMA-Small	1.21128	mfcc	vector_quant._default	cosine

Table 2. Results obtained with Music Similarity datasets.

dataset	MR	MRR	MDR	MAP	extractor	aggregator	distance
Covers80	41.575	0.19359	31.0	0.19359	chroma_stft	diff_stats_1	cosine_oti
YouTubeCovers	7.97143	0.6942	1.0	0.36114	pitch_cont._seg.	octave_abst.	lcs_circular_mean
Covers1000	144.041	0.25731	35.0	0.19159	pitch_cont._seg.	octave_abst.	lcs_circular_mean
Mazurkas	4.15724	0.95774	1.0	0.82286	pitch_cont._seg.	octave_abst.	levenshtein_circular_max
SHS9K	47.57883	0.40387	6.0	0.05102	pitch_cont._seg.	octave_abst.	lcs_circular_mean

Table 3. Results obtained with Cover Song Identification datasets.

The results of the second experiment are displayed as follows: the best models for each dataset are presented in Table 3, and the entire list of metrics computed in this experiment is published in `https://rppbodo.github.io/phd/experiment_2.html`.

4.4 Discussion

Analysing the models that achieved the best weighted mean IICSR for MIQ-QBSH, IOACAS-QBH, and Maria Panteli's melody dataset, it is possible to verify that all of them have Pitch Contour Segmentation as their feature, which matches the hypothesis that melodic features lead to better results for melodic datasets. Regarding the aggregators, two models have Octave Abstraction and one has Interval Abstraction. This is somehow expected, since the alternative abstractions (3-level and 5-level Pitch Contours) are relatively weaker due to their simplistic representations of the original pitch sequences.

The models that performed best for the Ballroom, MAST, and Maria Panteli's rhythm dataset are relatively surprising, not only because none of the features are specifically designed for rhythmic similarity tasks, but also because they are very different from each other: Spectral Bandwidth is related to the spectrum spread, MFCC is usually associated with timbre, and Chroma Energy Normalized Statistics (CENS) is a tonal feature.

Regarding the highest weighted mean IICSR obtained for GTZAN, 1517-Artists, and FMA-Small, two out of three best performing models have MFCC as their feature, and the other one has Spectral Contrast. MFCC is a feature usually related to timbre, so this matches the initial hypothesis. According to Jiang et al. [45], Spectral Contrast is

reported to have a better discriminating power for different music types than MFCC, so it is noteworthy that this feature has also emerged here.

The best models that lead to the lowest Mean Rank values for the five CSI datasets are shown in Table 3. Four models out of five share the same feature (Pitch Contour Segmentation) and the same aggregator (Octave Abstraction), which is a very good indication of the relevance of these methods, while the remaining model uses a Chromagram as feature. All features from the best models encode tonal information, which matches the observation in the literature that tonal differences are the less frequent between versions [22,46,47].

5 Conclusions

In this paper we introduced a modular music similarity framework designed to benchmark 690 music similarity models applied to specific music information retrieval tasks. Our experiments compared these models under several datasets compiled for tasks requiring different music similarity perspectives, showing that the choices of features, aggregators and distances not only have a significant impact on the performance of the corresponding models, but also that many useful techniques and combinations have been largely overlooked by the music similarity literature, corroborating the importance of comparative studies such as the present one.

As future work, we consider expanding the lists of features (HPCP, crema-PCP, Onset Patterns, Scale Transform, Pitch Bihistogram, Intervalgram, etc), aggregators (Dynamic Time Warping (DTW), Self-Organizing Map (SOM), vector quantization using tree-based clustering, n-grams, etc), and distances (Mahalanobis, Jensen-Shannon, Smith- Waterman, Mongeau-Sankoff, etc) in the music similarity framework, as well as incorporating alternative approaches to music similarity that not necessarily follow the current pipeline, such as feature learning [16,17], metric learning [18,19], and deep neural networks [20,21].

References

1. J Stephen Downie. The music information retrieval evaluation exchange (2005–2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.
2. Meinard Müller. *Fundamentals of music processing: Audio, analysis, algorithms, applications*. Springer, 2015.
3. Peter Knees and Markus Schedl. *Music Similarity and Retrieval: An Introduction to Audio- and Web-based Strategies*. Springer, 2016.
4. Jonathan T Foote. Content-based retrieval of music and audio. In *Multimedia Storage and Archiving Systems II*, volume 3229, pages 138–147. International Society for Optics and Photonics, 1997.
5. Tao Li and Mitsunori Ogihara. Content-based music similarity search and emotion detection. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 5, pages V–705. IEEE, 2004.
6. Klaus Seyerlehner, Markus Schedl, Tim Pohle, and Peter Knees. Using block-level features for genre classification, tag classification and music similarity estimation. *Submission to Audio Music Similarity and Retrieval Task of MIREX*, 2010, 2010.

7. Xiaoqing Yu, Jing Zhang, Junwei Liu, Wanggen Wan, and Wei Yang. An audio retrieval method based on chromagram and distance metrics. In *2010 International Conference on Audio, Language and Image Processing*, pages 425–428. IEEE, 2010.
8. Peter Knees and Markus Schedl. A survey of music similarity and recommendation from music context data. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 10(1):1–21, 2013.
9. Christopher J Tralie. Early mfcc and hpcp fusion for robust cover song identification. *arXiv preprint arXiv:1707.04680*, 2017.
10. Beth Logan and Ariel Salomon. A music similarity function based on signal analysis. In *ICME*, pages 22–25, 2001.
11. Jean-Julien Aucouturier, Francois Pachet, et al. Music similarity measures: What's the use? In *ISMIR*, pages 13–17, 2002.
12. Francois Pachet and Jean-Julien Aucouturier. Improving timbre similarity: How high is the sky. *Journal of negative results in speech and audio sciences*, 1(1):1–13, 2004.
13. Adam Berenzweig, Beth Logan, Daniel PW Ellis, and Brian Whitman. A large-scale evaluation of acoustic and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.
14. Rainer Typke, Frans Wiering, and Remco C Veltkamp. Evaluating the earth mover's distance for measuring symbolic melodic similarity. In *MIREX-ISMIR 2005: 6th International Conference on Music Information Retrieval*. Citeseer, 2005.
15. Dominik Schnitzer, Arthur Flexer, and Gerhard Widmer. A fast audio similarity retrieval method for millions of music tracks. *Multimedia Tools and Applications*, 58(1):23–40, 2012.
16. Maria Panteli, Emmanouil Benetos, Simon Dixon, et al. Learning a feature space for similarity in world music. ISMIR, 2016.
17. Zhesong Yu, Xiaoshuo Xu, Xiaoou Chen, and Deshun Yang. Learning a representation for cover song identification using convolutional neural network. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 541–545. IEEE, 2020.
18. Malcolm Slaney, Kilian Weinberger, and William White. Learning a metric for music similarity. In *International Symposium on Music Information Retrieval (ISMIR)*, volume 148, 2008.
19. Hoon Heo, Hyunwoo J Kim, Wan Soo Kim, and Kyogu Lee. Cover song identification with metric learning using distance as a feature. In *ISMIR*, pages 628–634, 2017.
20. Manan Mehta, Anmol Sajnani, and Radhika Chapaneri. Cover song identification with pairwise cross-similarity matrix using deep learning. In *2019 IEEE Bombay Section Signature Conference (IBSSC)*, pages 1–5. IEEE, 2019.
21. Mohamadreza Sheikh Fathollahi and Farbod Razzazi. Music similarity measurement and recommendation system using convolutional neural networks. *International Journal of Multimedia Information Retrieval*, 10(1):43–53, 2021.
22. Joan Serra, Emilia Gomez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. pages 307–332, 2010.
23. Jesper Hojvang Jensen, Mads G Christensen, Daniel PW Ellis, and Soren Holdt Jensen. A tempo-insensitive distance measure for cover song identification based on chroma features. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 2209–2212. IEEE, 2008.
24. Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.
25. Suman Ravuri and Daniel PW Ellis. Cover song detection: from high scores to general classification. In *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 65–68. IEEE, 2010.

26. Dmitri Tymoczko. Three conceptions of musical distance. In *International Conference on Mathematics and Computation in Music*, pages 258–272. Springer, 2009.
27. Asif Ghias, Jonathan Logan, David Chamberlin, and Brian C Smith. Query by humming: Musical information retrieval in an audio database. In *Proceedings of the third ACM international conference on Multimedia*, pages 231–236, 1995.
28. Wei-Ho Tsai, Hung-Ming Yu, Hsin-Min Wang, et al. Query-by-example technique for retrieving cover versions of popular songs with similar melodies. In *ISMIR*, volume 5, pages 183–190, 2005.
29. Klaus Frieler and Daniel Müllensiefen. The simile algorithm for melodic similarity. *Proceedings of the Annual Music Information Retrieval Evaluation exchange*, 2005.
30. Matija Marolt. A mid-level melody-based representation for calculating audio similarity. In *ISMIR*, pages 280–285, 2006.
31. Seungmin Rho and Eenjun Hwang. Fmf: Query adaptive melody retrieval system. *Journal of Systems and Software*, 79(1):43–56, 2006.
32. Justin Salamon, Joan Serra, and Emilia Gómez. Tonal representations for music retrieval: from version identification to query-by-humming. *International Journal of Multimedia Information Retrieval*, 2(1):45–58, 2013.
33. Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based organization and visualization of music archives. In *Proceedings of the tenth ACM international conference on Multimedia*, pages 570–579, 2002.
34. Costas Panagiotakis and Georgios Tziritas. A speech/music discriminator based on rms and zero-crossings. *IEEE Transactions on multimedia*, 7(1):155–166, 2005.
35. Cory McKay and I Fujinaga. Automatic music classification and similarity analysis. In *International Conference on Music Information Retrieval*. Citeseer, 2005.
36. Holger H Hoos, Kai Renz, and Marko Görg. Guido/mir-an experimental musical information retrieval system based on guido music notation. In *ISMIR*, pages 41–50, 2001.
37. Alexandra Uitdenbogerd and Justin Zobel. Melodic matching techniques for large music databases. In *Proceedings of the seventh ACM international conference on Multimedia (Part 1)*, pages 57–66, 1999.
38. Matthew Kelly. *Evaluation of melody similarity measures*. PhD thesis, 2012.
39. Kjell Lemström and Esko Ukkonen. Including interval encoding into edit distance based music comparison and retrieval. In *Proc. AISB*, pages 53–60. Citeseer, 2000.
40. Marc Sarfati, Anthony Hu, and Jonathan Donier. Ensemble-based cover song detection. *arXiv preprint arXiv:1905.11700*, 2019.
41. J Stephen Downie, Mert Bay, Andreas F Ehmann, and M Cameron Jones. Audio cover song identification: Mirex 2006-2007 results and analyses. In *ISMIR*, pages 468–474, 2008.
42. Kim Falk. *Practical recommender systems*. Manning Publications, 2019.
43. George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals. In *Proceedings of the 2nd international symposium on music information retrieval, Indiana*, 2001.
44. Jean-Julien Aucouturier and Francois Pachet. Representing musical genre: A state of the art. *Journal of new music research*, 32(1):83–93, 2003.
45. Dan-Ning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proceedings. IEEE International Conference on Multimedia and Expo*, volume 1, pages 113–116. IEEE, 2002.
46. Justin Salamon, Joan Serrá, and Emilia Gómez. Melody, bass line, and harmony representations for music version identification. In *Proceedings of the 21st International Conference on World Wide Web*, pages 887–894, 2012.
47. Ning Chen, Mingyu Li, and Haidong Xiao. Two-layer similarity fusion model for cover song identification. *EURASIP Journal on Audio, Speech, and Music Processing*, 2017(1):1–15, 2017.

Deep Learning-Based Music Instrument Recognition: Exploring Learned Feature Representations

Michael Taenzer*, Stylianos I. Mimilakis*, and Jakob Abeßer

Fraunhofer IDMT, Semantic Music Technologies Group, Ilmenau, Germany
{tzr, mis, abr}@idmt.fraunhofer.de

Abstract. In this work, we focus on the problem of automatic instrument recognition (AIR) using supervised learning. In particular, we follow a state-of-the-art AIR approach that combines a deep convolutional neural network (CNN) architecture with an attention mechanism. This attention mechanism is conditioned on a learned input feature representation, which itself is extracted by another CNN model acting as a feature extractor. The extractor is pre-trained on a large-scale audio dataset using discriminative objectives for sound event detection. In our experiments, we show that when using log-mel spectrograms as input features instead, the performance of the CNN-based AIR algorithm decreases significantly. Hence, our results indicate that the feature representations are the main factor that affects the performance of the AIR algorithm. Furthermore, we show that various pre-training tasks affect the AIR performance in different ways for subsets of the music instrument classes.

Keywords: music instrument recognition, deep learning, representation learning

1 Introduction

Real world music recordings often consist of multiple music instruments that can be active simultaneously. Detecting individual instruments or instrument families is an important research problem in areas such as machine listening, music information retrieval (MIR), and (music) source separation. The problem of detecting and categorizing the active instruments is often referred to as automatic instrument recognition (AIR). Recent approaches to AIR are mostly based on deep convolutional neural networks (CNNs) [1–5].

One commonality in deep learning approaches for AIR is that they consist of three modules [1, 3–5], namely the pre-processing, embedding, and classification modules. The first module pre-processes and transforms the respective input music waveform into a compact signal representation. The most common transforms are the short-time Fourier transform (STFT) and related filter-banks such as Mel-bands [1, 3, 4, 6] and the constant-Q transform (CQT) [7]. Common operations as pre-processing steps are harmonic and percussive separation [3] as well as logarithmic magnitude compression and data normalization or standardization [4].

* Equally contributing authors

The second module, referred to as embedding, accepts as input the pre-processed and transformed music waveform from the first module. It yields a feature *representation* that is used to condition the last module, i. e., the classification module, which is responsible for computing the posterior, i.e., the label probability, of the corresponding instrument classes (e.g., "electric guitar" or "piano"). Most often, the embedding and classification modules are learned jointly during a training procedure that is based on supervised learning, in which the class labels for each recording are given from a curated dataset [1, 4].

Regarding the embedding module, a common ingredient in the related literature is the usage of CNNs [1, 3, 4] and, more recently, CNN-based attention mechanisms [5, 8]. The approaches employing attention mechanisms are experimentally shown to yield state-of-the-art results, and it is assumed that the attention mechanism is responsible for the success of the methods. However, the studies presented in [5] and [8] condition the attention mechanism on a feature representation that is computed using a *pre-trained* CNN: That CNN, in particular the VGGish network [9], is initially trained for audio event detection (AED) in a supervised way using general audio signals and classes obtained from Audio Set [10], before being applied on the task of AIR. This means that the attention-based approaches to AIR make use of transfer learning [11, 12]. This differs from other approaches, which learn the representations jointly for the task of AIR [1, 3, 4]. Therefore, it could be argued that the observed increase in performance of such attention-based models rather needs to be attributed to the discriminative power of the feature representations from the CNN, which was previously learnt from more general audio signals instead of solely music signals [13].

In this work, we analyze the impact of the role of learning feature representations for an attention mechanism for music instrument classification performance. It should be noted that it is not our intention to conduct a comparative study on attention mechanisms versus representation learning, as we believe that both are equally beneficial for the task at hand. Instead, we aim to show that deep learning approaches to AIR can substantially benefit from employing representations that are learned using reconstruction or alignment optimization objectives [14] as well as datasets that contain general purpose audio signals [10].

To answer our research question, we focus on the attention-based model presented in [5], which is trained and tested on the respective subsets of the OpenMIC dataset [15]. To investigate the influence of different feature representations, we experiment with various commonly used filter-banks, such as (log) Mel-spectrograms, and learned representations. For the latter case, different datasets and optimization objectives are used to pre-train the CNN responsible for yielding the feature representations. These are described in sections 3 and 4.1, respectively.

2 Attention-based Model

The attention-based model for AIR from the work presented in [5] is illustrated in Fig. 1 embedded into our general experimental pipeline as described in the following.

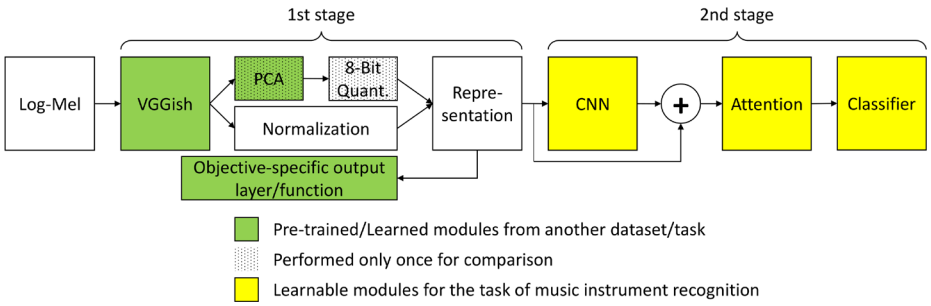


Fig. 1. An illustration of our experimental pipeline and the method presented in [5] that employs an attention mechanism and a pre-trained CNN (VGGish) for computing the feature representation(s).

2.1 Input pre-processing

An input time-domain music signal is first re-sampled at a sampling frequency of 16 kHz and then transformed into a time-frequency representation using the short-time Fourier transform (STFT). The parameters for the STFT are a window size of 25 ms using the Hann function and a hop-size of 10 ms. Each windowed segment is zero-padded to 512 samples. From the magnitude of the STFT, a Mel-spectrogram with 64 mel bands is computed. We apply log-magnitude scaling to the Mel-spectrogram, yielding a final input spectral representation denoted as "Log-Mel" in Fig. 1.

2.2 Post-processing & Representation

The Log-Mel is used to condition the VGGish network presented in [16]. This network comprises six convolutional (conv) blocks followed by three fully-connected feed-forward (dense) layers (FC). Each conv block consists of a two-dimensional conv layer (2Dconv), the rectified linear unit (ReLU) activation function, and a two-dimensional max-pooling operation. The numbers of kernels across the conv blocks are $\{64, 128, 256, 256, 512, 512\}$. The kernel sizes for the conv and max-pooling operations in all conv blocks are 3×3 and 2×2, respectively, and the stride size is set to 1. Furthermore, zero-padding is applied to preserve the size of the intermediate latent representations (activation maps), which are computed using the convolutions.

The outputs of each kernel in the last conv block are concatenated to a vector and then given as input to the first FC. The number of output units in each FC is $\{4096, 4096, 128\}$, respectively. The ReLU activation function is used after each FC. The output of the VGGish is a feature representation that summarizes approximately one second of spectral information into a single embedding vector [16].

This output representation is then post-processed by applying a whitening transform using principal component analysis (PCA). The bases for the PCA are *pre-computed* from the audio signals' corresponding representation obtained by the VGGish [15, 16] using the training subset of Audio Set [10]. This whitened feature representation is 8-bit quantized and mapped to the range of $[0, 1]$.

2.3 Additional CNN, Attention Mechanism & Classification

The above representation is processed by a block of 2D CNNs, which precede the attention module. It consists of three 2Dconv layers with unit stride and a group-normalization layer. Each layer employs 128 1×1-kernels. The output of the group-normalization layer is then updated by means of residual connections using the information of the post-processed representation.

The output of the residual connections is given to the attention module that consists of two 2Dconv layers with kernel size 1×1. The number of kernels in each 2Dconv layer is equal to the number of classes. The representation is fed to each 2Dconv layer in the attention module, followed by the application of the element-wise sigmoid activation function. The output of the conv layer responsible for decoding the attention embedding is normalized to unit sum with respect to the time-frame information. The output of the conv layer responsible for the class activity is used to gate the normalized output of the other conv layer.

Using this attention mechanism, the posterior can be computed by aggregating the time-information of the output of the attention mechanism, followed by the application of the hard-tanh function linear in the range of $[0, 1]$, equal to 1 for values > 1, and 0 for negative values. The aggregation is performed due to the weakly annotated labels contained in OpenMIC [5].

3 Datasets

To optimize the overall model parameters contained in each module described in Section 2, a two-stage training scheme is employed. In the first stage, the modules that are used to compute the feature representation, i. e., as illustrated in green color in Fig. 1, are pre-trained on a task different from AIR. The second stage uses the pre-trained modules from the previous stage, and optimizes the rest of the modules, i. e., the yellow modules illustrated in Fig. 1, using the labels associated with the task of AIR.

Table 1. Usage of different datasets for the respective training stages and objectives. Denoising from additive noise is indicated by $+$, and from multiplicative noise as $\odot$.

Datasets		
1st stage	**2nd stage**	**1st stage objectives**
Audio Set	OpenMIC	General purpose AED
NSynth	OpenMIC	Textual description, Denoising $+/\odot$
Freesound	OpenMIC	Tag Alignment

For the first stage, we employ the already optimized VGGish embeddings [15][1] pre-trained on the Audio Set [10], and the NSynth [17] and Freesound [18] datasets. Depending on the dataset for this stage, various pre-training objectives are used (see Section 4.1). For the second stage, we utilize only OpenMIC [15] for both training and testing, with the respective subsets used in [5]. Table 1 provides an overview of this.

[1] Publicly available under `https://github.com/cosmir/openmic-2018`

4 Experimental Procedure

Since the PCA and 8-bit quantization steps in the post-processing of the feature representation from the VGGish are irrelevant to the scope of our work, we have excluded them from our experiments. Instead, a simple normalization to $[0, 1]$ is applied to the representation during the second stage of training to avoid any crucial performance discrepancies due to the inductive biases of the attention-based method for AIR.

4.1 Pre-training Objectives (First Stage)

This section provides technical details regarding the experimental setup for each employed objective in the first training stage for optimizing the parameters of the VGGish. Table 1 gives a summary and overview. For all pre-training learning objectives, the Adam optimizer is used with a fixed learning rate of $1e^{-4}$. Furthermore, the batch size is set to 64 and an early stopping mechanism is used, which terminates the training procedure if the used criterion (validation loss) has not improved for five consecutive iterations throughout the entire training data. The maximum number of training epochs is 50. All parameters are initialized randomly with samples drawn from a uniform distribution and scaled using the method presented in [19].

Textual description One investigated pre-training objective is the prediction of the textual description of a music recording. This objective draws inspiration from the field of audio captioning, which aims at generating a textual description of an audio signal. Subject to the goal of this work, we employ the NSynth dataset [17] that contains the following textual descriptions of the musical notes for every recording in the dataset: {'bright', 'dark', 'distortion', 'decay', 'presence', 'multiphonic', 'modulation', 'percussive', 'reverb', 'rhythmic'}[2]. For using the textual descriptions of the music files to train the VGGish, we employ the Word2Vec language model, presented in [20] and pre-trained on an English vocabulary, to yield a vector representation of each description.

The Word2Vec model encodes each input word, in our case the textual description, into a 300-dimensional vector embedding. That vector is used as the target to learn the parameters of the VGGish. To do so, the output of the VGGish is given as input to a trainable batch-norm layer and two fully-connected feed-forward (dense) layers (FC). The first FC employs the non-linear tanh activation function, whereas the second does not employ any element-wise non-linear functions. The number of units in the FCs is set to 300. The VGGish network applied on an audio example from the NSynth dataset yields a single vector, because every NSynth example has a length of one second. Therefore, it is not necessary to aggregate over temporal information. The parameters of the VGGish and the following block of batch-norm and FCs are jointly optimized by minimizing the cosine loss between the predicted and target word-vector embeddings. The margin hyperparameter in the computation of the cosine loss is set to 0.5.

[2] We replaced the original NSynth descriptions {'fast decay', 'long release', 'nonlinear env' 'tempo-synced'} with {'decay', presence', 'modulation' 'rhythmic'}, based on the additional description contained in the dataset. This was due to the fact that the original descriptions could not be fully encoded by the employed language model.

Signal Recovery Another training objective we investigate is the recovery of the original Log-Mel spectrogram from a corrupted version of the signal. The goal of this objective is to enforce the representation from the VGGish to encode the relevant information contained in the Log-Mel. To do so, we employ denoising auto-encoders (DAEs) [21] and corrupt the Log-Mel in two different ways before it is input to the VGGish: a) with element-wise additive noise ($+$) drawn from a Gaussian distribution with zero mean and 0.1 standard deviation, and b) with element-wise multiplicative noise ($\odot$). For the latter case, random values are drawn from a Bernoulli distribution with $p = 0.5$.

To decode the representation of the corrupted Log-Mel obtained by the VGGish, we employ a single block of conv layers containing four transposed one-dimensional conv (1Dconv) layers. We use transposed 1Dconv layers to be able to recover (upsample back to) the original dimensionality regarding time-frames, which the VGGish reduced. The number of kernels and their size in each layer are $\{128, 64, 64, 64\}$ and $\{10, 21, 31, 37\}$, respectively. No zero-padding is applied between each convolution. Furthermore, the first three 1Dconv layers employ the leaky ReLU activation function with a leaky-factor of $\{0.1, 0.25, 0.5\}$, respectively. The last conv layer uses a linear activation function. These hyperparameters are chosen experimentally so that a reasonable convergence is achieved using a random and smaller subset of NSynth.

Audio & Tag Alignment We also explore the objective of aligning audio and associated tags. The alignment is achieved by maximizing the agreement of the computed audio and tag representations using a contrastive loss. We employ the tag encoder and the corresponding hyperparameters following the method presented in [14], whose goal is to compute representations that reflect acoustic and semantic characteristics of audio signals. For the audio encoder, we use the VGGish as discussed above. To match the dimensionality used by the audio tag encoder, we apply an affine transformation after the VGGish. The optimization hyperparameters for this configuration are taken from [14].

4.2 Downstream Instrument Recognition (Second Stage)

After optimizing the parameters of the VGGish with one of the above pre-training objectives, the VGGish computes the representations of the audio files contained in OpenMIC. Together with the corresponding labels within OpenMIC, these are then used to optimize the parameters of the CNN and the attention mechanism. To that aim, we use the existing splits of OpenMIC for training and validation as employed in [5].

For training, we use the binary cross-entropy loss function. The hyperparameters for optimization are the Adam algorithm with a learning rate of $5e^{-4}$, a batch size equal to 128 data points and a total number of 350 training epochs, following [5]. After every iteration over the entire training subset, we evaluate the model performance on the validation subset. After training, we select the best set of parameters based on the obtained evaluation score calculated in every iteration.

5 Evaluation

While the total number of audio files per instrument in OpenMIC is balanced, the number of positive and negative examples varies from one instrument class to another. For

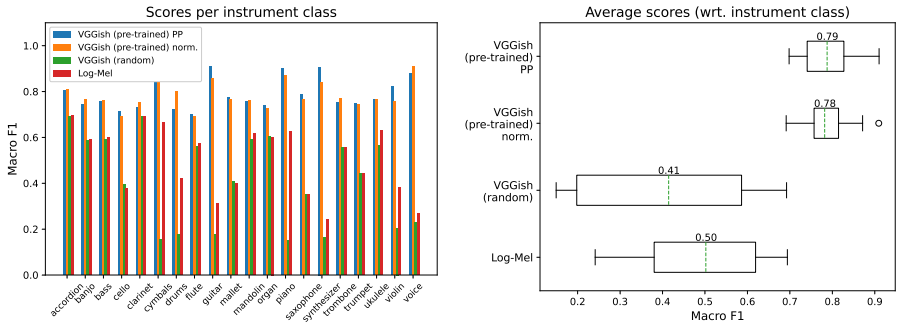


Fig. 2. F1-scores per instrument class (left) and overall (right) for the attention-based model [5], conditioned on feature representations computed using the pre-trained VGGish and a randomly initialized VGGish, and the Log-Mel representation. Note the marginal differences between post-processing (PP) and normalization (norm.) For VGGish (pre-trained), the Audio Set is used in the first training stage. For VGGish (random) and Log-Mel, no first training stage is performed.

this reason, we compute the macro-average F1-score (F1-macro) explicitly for positive and negative classes of every instrument class to evaluate both the parameters of the attention-based model and the pre-training stages, during both training and validation phases. During evaluation, the outputs of the classifier are subject to a post-processing operation that thresholds to zero values below 0.5 and unity values otherwise. Finally, to determine the benefits of each objective, we test the attention mechanism each time it has been trained with a different feature representation on the test subset of OpenMIC.

6 Results & Discussion

6.1 Representation Post-processing: Impact on performance

First, we examine the impact of the post-processing steps (see Section 2.2) versus normalization on classification performance, illustrated in Fig. 2. From the F-score, it can be seen that a simple scaling of the feature representation induces only a marginal performance drop. This allows us to omit further data-dependent post-processing stages that are irrelevant to our research question, yet might impose some performance discrepancies. From the barplot it can also be observed that without the PCA and quantization steps the performance increases marginally for the banjo, clarinet, drums, mandolin, trombone, and voice instrument classes.

6.2 Learned Representations: Impact on Performance

To highlight the impact of the learned representations on the performance for classifying musical instruments using the discussed attention mechanism, Fig. 2 shows the results from the attention-based model conditioned on three feature representations. These are computed from the pre-trained VGGish, a randomly initialized VGGish, and using the Log-Mel representation directly, i. e., the VGGish acts as an identity operator.

The Fig. 2 boxplot highlights two observations: First, regarding F1-macro, the discriminative power of the pre-trained VGGish is responsible for obtaining the best classification performance. Secondly, even an unoptimized (randomly initialized) VGGish can be used to compute a feature representation that yields a classification performance comparable to Log-Mel, which is a common feature representation for audio classification tasks. However, the barplot demonstrates that Log-Mel outperforms the representation from the randomly initialized VGGish for a few musical instruments classes including cymbals, ukulele, violin, and voice. These two tendencies suggest that the performance of the attention-based model may be accredited mostly to the discriminative power of the representation yielded by the VGGish. They also imply that different objectives or datasets may be used to pre-train the VGGish and yield different results.

Fig. 3 explores this observed direction with the results of the classification performance using the various objectives described in Section 4.1. As can be seen in the boxplot, the pre-training tasks of signal recovery and textual description provide significant improvements by 0.11 in the F-score over the randomly initialized VGGish. Compared to the Log-Mel features, marginal improvements of approximately 0.02 are observed. The barplot shows that each pre-training objective seems to be beneficial for different musical instrument classes. For example, Textual provides competitive results for the instrument classes mallet, mandolin, organ, ukulele, and voice, while Den $\odot$ seems to work well for piano, ukulele, and violin. Den $+$ provides improvements for percussive musical instruments such as cymbals and drums. In any case, the VGGish pre-trained on Audio Set (see Fig. 2) significantly outperforms the best performing models which employ NSynth.

Plausible explanations for these observed discrepancies lie in the amount of data and variability within Audio Set, and in the naiveness (in the sense of simple and not carefully devised) of the pre-training objectives, e.g. the signal recovery. This explanation is underlined when considering the results for the Align objective (Fig. 3), which employs Freesound and uses a more sophisticated mechanism to exploit the information of the audio tags.

Fig. 3 shows that the usage of larger corpora in conjunction with a more sophisticated objective (Align) can lead to significant improvements in attention-based AIR compared to the signal recovery and textual description objectives. Nonetheless, it is still sub-optimal compared to VGGish pre-trained on Audio Set. The discrepancy in performance between the two may be accredited to the data availability. However, this finding highlights the trend that using more general purpose audio datasets can improve the downstream task of AIR.

7 Conclusions

In this work, we investigated the importance of the role of learning representations w.r.t. an attention mechanism in music instrument classification algorithms. To that aim, we focused on the attention-based model for music instrument recognition presented in [5], and experimentally explored the impact of various feature representations on the performance of the attention-based model. Our experimental findings highlight the following trends: i) Discriminative objectives in conjunction with large scale and general purpose

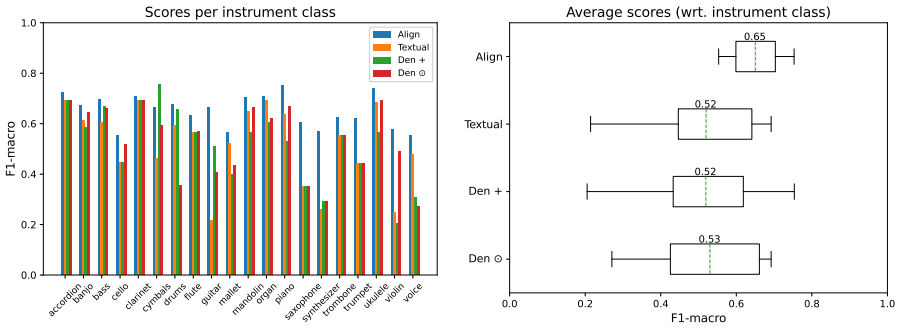


Fig. 3. F1-scores per instrument class (left) and overall (right) for the attention-based model [5], conditioned on feature representations computed from pre-training the VGGish for audio and tag alignment (Align), textual description (Textual), and signal recovery from additive noise (Den $+$) and multiplicative noise (Den $\odot$). The first training stage uses Freesound for Align, and NSynth for Textual, Den $+$ and Den $\odot$.

audio corpora are an important factor to be considered in AIR apart from the attention mechanism, ii) the usage of audio tags for computing representations is an attractive objective that yields competing performance, and iii) training objectives that take advantage of general audio and annotations or, respectively, the exploitation of multimodalities in a self-supervised manner are emerging directions for future research.

Acknowledgements This work has been supported by the German Research Foundation (AB 675/2-1).

References

1. Y. Han, J. Kim, and K. Lee. Deep Convolutional Neural Networks for Predominant Instrument Recognition in Polyphonic Music. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 25(1):208–221, 2017.
2. T. Park and T. Lee. Musical Instrument Sound Classification with Deep Convolutional Neural Network Using Feature Fusion Approach. *arXiv preprint arXiv:1512.07370*, 2015.
3. J. Gomez, J. Abeßer, and E. Cano. Jazz Solo Instrument Classification with Convolutional Neural Networks, Source Separation, and Transfer Learning. In *Proceedings of the 19th International Society of Music Information Retrieval Conference (ISMIR)*, pages 577–584, Paris, France, 2018.
4. M. Taenzer, J. Abeßer, S. I. Mimilakis, C. Weiß, M. Müller, and H. Lukashevich. Investigating CNN-Based Instrument Family Recognition for Western Classical Music Recordings. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 612–619, Delft, The Netherlands, 2019.
5. S. Gururani, M. Sharma, and A. Lerch. An Attention mechanism for musical instrument recognition. In *Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR)*, pages 83–90, Delft, The Netherlands, 2019.

6. S. I. Mimilakis, C. Weiß, V. Arifi-Müller, J. Abeßer, and M. Müller. Cross-Version Singing Voice Detection in Opera Recordings: Challenges for Supervised Learning. In *Proceedings of the 12th International Workshop on Machine Learning and Music (MML)*, pages 429–436, Würzburg, Germany, 2019.
7. Y.-N. Hung and Y.-H. Yang. Frame-Level Instrument Recognition by Timbre and Pitch. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–142, Paris, France, 2018.
8. K. Watcharasupat, S. Gururani, and A. Lerch. Visual Attention for Musical Instrument Recognition. *arXiv preprint arXiv:2006.09640*, 2020.
9. S. Hershey, S. Chaudhuri, D. P. W. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, M. Slaney, R. J. Weiss, and K. Wilson. CNN architectures for large-scale audio classification. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, New Orleans, LA, USA, 2017.
10. J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter. Audio Set: An Ontology and Human-Labeled Dataset for Audio Events. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017.
11. M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1717–1724, 2014.
12. M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning Transferable Features with Deep Adaptation Networks. In *Proceedings of the 32nd International Conference on Machine Learning (ICML) - Volume 37*, pages 97–105, Lille, France, 2015.
13. Y. Bengio, A. Courville, and P. Vincent. Representation Learning: A Review and New Perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, Aug. 2013.
14. X. Favory, K. Drossos, T. Virtanen, and X. Serra. Coala: Co-aligned autoencoders for learning semantically enriched audio representations. *arXiv preprint arXiv:2006.08386*, 2020.
15. E. J. Humphrey, S. Durand, and B. Mcfee. OpenMIC-2018: An Open Data-set for Multiple Instrument Recognition. In *Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR)*, pages 438–444, Paris, France, 2018.
16. A. Jansen, J. F. Gemmeke, D. P. W. Ellis, X. Liu, W. Lawrence, and D. Freedman. Large-Scale Audio Event Discovery in One Million YouTube Videos. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 786–790, 2017.
17. J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi. Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. *arXiv preprint arXiv:1704.01279*, 2017.
18. F. Font, G. Roma, and X. Serra. Freesound technical demo. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 411–412, New York, NY, USA, 2013.
19. X. Glorot and Y. Bengio. Understanding the Difficulty of Training Deep Feedforward Neural Networks. In *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*, pages 249–256, 2010.
20. J. Pennington, R. Socher, and C. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Oct. 2014.
21. P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol. Extracting and Composing Robust Features with Denoising Autoencoders. In *Proceedings of the 25th International Conference on Machine Learning (ICML)*, pages 1096–1103, Helsinki, Finland, 2008. ACM.

Hierarchical Predictive Coding and Interpretable Audio Analysis-Synthesis

André Ofner, Johannes Schleiss and Sebastian Stober

Otto von Guericke University, Magdeburg, Germany
{ofner,schleiss,stober}@ovgu.de

Abstract. Humans efficiently extract relevant information from complex auditory stimuli. Oftentimes, the interpretation of the signal is ambiguous and musical meaning is derived from the subjective context. Predictive processing interpretations of brain function describe subjective music experience driven by hierarchical precision-weighted expectations. There is still a lack of efficient and structurally interpretable machine learning models operating on audio featuring such biological plausibility. We therefore propose a bio-plausible predictive coding model that analyses auditory signals in comparison to a continuously updated differentiable generative model. For this, we discuss and build upon the connections between Infinite Impulse Response filters, Kalman filters, and the inference in predictive coding with local update rules. Our results show that such gradient-based predictive coding is useful for classical digital signal processing applications like audio filtering. We test the model capability on beat tracking and audio filtering tasks and conclude by showing how top-down expectations modulate the activity on lower layers during prediction.

Keywords: Predictive Processing, Machine learning, Digital Signal Processing

1 Introduction

1.1 Audio Processing and Predictive Coding in the Human Brain

Research on human auditory processing has demonstrated that humans are efficient at tracking stochastic auditory regularities and can even disentangle stationary parts, e.g. fundamental frequencies, from dynamic transformations, e.g. resonances, in musical events. The predictive coding (PC) theory is a popular framework in neuroscience that explains how such complex human processing could arise from a relatively simple repeated algorithmic pattern implemented in neurons, namely the reduction of prediction errors [1, 2]. Recent advances in machine learning have progressed towards predictive coding models that update simulated neurons with errors computed local to these neurons, in contrast to the backpropagation through entire neural networks that drive most current deep learning systems [3]. Through the use of local errors and simple neural operations (e.g. summation or addition) PC networks are plausible models of the computations in biological neurons. From an engineering perspective, predictive coding networks (PCN) with a single layer already deliver useful computations, like the source-filter separation in Linear Predictive Coding (LPC), a widely used Digital Signal Processing (DSP) method. To live up to their full potential, PCNs need hierarchical structure. In hierarchical PCNs hidden layers predict the expected latent states of lower layers. However, there is still a

lack of hierarchical and biologically plausible machine learning models that combine the possibility to operate on raw audio with reasonable performance on classical DSP tasks. These tasks can include audio filtering or extracting musical information, e.g. beat timings, from audio.

1.2 Hierarchical Predictive Coding and Digital Signal Processing

The number of existing studies employing predictive coding to process raw audio is limited and available methods are generally difficult to interpret. Moreover, PC models in neuroscience are generally restricted to simple auditory stimuli or even symbolic inputs [4, 5]. Still, there are similarities between the structures of Infinite Impulse Response (IIR) filters and recurrent neural networks (RNN), classes that are already widely used in DSP applications and those models that model human (auditory) cognition more specifically, in particular the Kalman filter or predictive coding networks. These connections will be discussed in more detail in Section 2.

A major challenge when employing predictive coding networks for engineering tasks is that they only deliver approximate results during learning and inference. This poses a major drawback in the context of DSP tasks, where high accuracy is generally required. Furthermore, it is difficult to design efficiently operating hierarchical PC models, which would have the advantage of naturally scaling to larger DSP systems with meaningful cognitive interpretations. To solve these challenges, we resort to the structural similarities between PC models and established DSP methods in the next section and then introduce a hierarchical PC model [1].

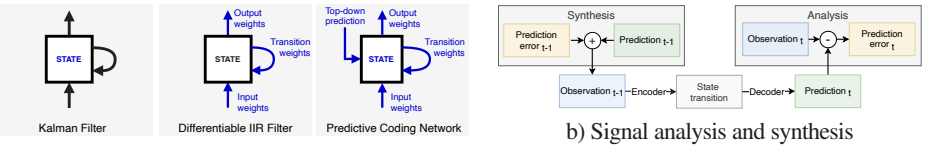
2 Related Work

The similarity between IIR filters, Kalman filters, RNNs, and predictive coding networks is particularly apparent when one views these models in their state-space (SSM) form. Figure 1 a) provides an overview of these related classes in state-space form, such as they are used in tasks typical for each class. Aspects of learned model structure, such as filter coefficients, are referred to as weights in the context of artificial networks. Generally speaking, ”inference” refers to employing these given coefficients (i.e. weights) to update hidden representations, while ”learning” refers to the slower process of optimizing weights.

While the signal flow of the model classes is directly comparable, differences arise in the way inference and learning are addressed in typical tasks. Kalman filters are usually used for dynamic inference given prior assumptions on the data, resulting in mathematically exact updates of their latent state. The deterministic class of IIR filters is typically used to apply a previously designed transfer function to incoming signals, where output signals are a weighted combination of previously processed signals. Some exceptions, such as differentiable IIR filters allow to learn weights during application [6]. Kalman filters and predictive coding networks are typically modeled as probabilistic generative models, keeping track of an inferred latent state with associated variance (or inverse precision). Both have found applications in modeling cognitive and neural processes. In contrast to Kalman filters, optimization in predictive coding networks generally addresses state inference and weights learning simultaneously.

Finally, PCNs can include internal predictions of their latent states, i.e. ”top-down” expectations about activities in lower PCN layers [2, 7]. This hierarchical structure is similar,

[1] Code is available at github.com/andreofner/APC



a) Comparison in state-space form

b) Signal analysis and synthesis with autoregressive predictive coding

Fig. 1. a) Comparison of Kalman filters, differentiable IIR filters, and gradient-based predictive coding networks in state-space form. Blue color indicates variables that are optimized in a typical filtering application for each model. b) Signal analysis and synthesis with autoregressive predictive coding and linear activation functions: In the analysis stage, observations at time-step t are mapped to hidden states using encoder weights. The learned transition dynamics are then applied to the latent state. Outgoing predictions for the next timestep $t+1$ are computed via decoder weights that map from the updated latent state to the expected sensory input. During synthesis, the prediction error is fed to the model jointly with the previous prediction.

but not identical, to the multi-layer architecture of deep neural networks, which typically lack the feedback connections that are inherent to PCNs. More specifically, DNNs can be interpreted as corresponding to pyramidal dendritic connections in the biological counterpart. This means that DNNs, possibly with multiple layers, connect adjacent variables in PCN layers [8]. Finally, existing work on PCN architectures has explored "dynamical" predictive coding, where not only the activity of lower layers but also (multiple) temporal derivatives are modelled [9]. Here, we explore the audio DSP capabilities of single-layer and hierarchical PCN models interpreted as biologically plausible Neural Kalman filters. This PCN class has been discussed for single-layer models in [10].

2.1 Autoregressive Signal Filtering with State-Space Models

Signal analysis with autoregressive filters at discrete time-steps t can be described with respect to a steady state transfer function $H(z)$

$$H(z) = \frac{G}{1 - \sum_{j=1}^{k} a_j z^{-j}} = \frac{G}{A(z)} \tag{1}$$

with input gain G [11, 12]. The parameters a_j with $1 \leq j \leq k$ and G of this state transfer function can be optimized with respect to the prediction error $e(x)$ between predicted signal $p(t)$ and observed signal $o(t)$, also referred to as excitation or residual signal:

$$e(t) = \frac{1}{G}(o(t) - \sum_{j=1}^{k} a_j o(t-j)) \tag{2}$$

The SSM of this generalized prediction error filter is updated with the following difference equation:

$$\begin{aligned} z[t+1] &= A[t]z[t] \\ o[t] &= C[t]z[t] \end{aligned} \tag{3}$$

where $z[t]$ is the state vector at timestep t and the prediction coefficients a_j are represented by weights A and C. All four discussed model classes, despite originating from the different

fields can be interpreted in prediction error minimizing SSM form. Linear predictive coding (LPC), a widely used DSP tool, draws from this possibility for the design of IIR coefficients. LPC is typically used for signal compression, particularly for speech coding, by separating stationary residual signals from imposed resonances [13]. This theoretically allows to analyse and synthesize signals using the same model. However, the efficient algorithms employed in LPC are not directly biologically interpretable and generally do not actually use a SSM to find the coefficients. From this perspective, our work generalises LPC towards the more general class of hierarchical PCN, where analysis and synthesis use the same model.

RNN and Differentiable IIR Filter Recurrent neural networks, in their simplest form, can be expressed by the following difference equations [6, 14]:

$$\begin{aligned} z[t+1] &= \sigma_z(W_z z[t] + U_z x[t+1] + b_z) \\ y[t+1] &= \sigma_y(W_y z[t+1] + b_y) \end{aligned} \tag{4}$$

with hidden states z, inputs x and outputs y. W and U are trainable weights and b are biases. Known from previous work is that, in the case where activation functions σ are (non-)linear and the biases are set to zero, this structure directly resembles a (non-)linear all-pole IIR filter

$$\begin{aligned} z[t+1] &= W_z z[t] + U_z x[t+1] \\ y[t+1] &= W_y z[t+1] \end{aligned} \tag{5}$$

which scales to arbitrary order of transfer functions $H(z)$ (also referred to as the filter order) and allows to train differentiable IIR filters using the optimization methodology for RNNs [6]. A useful generalized state space form for such IIR filters is

$$\begin{aligned} z[t+1] &= Az[t] + Bx[t] \\ y[t+1] &= Cz[t+1] + Dx[t+1] \end{aligned} \tag{6}$$

where matrices A,C represent the learnable weights for latent state transition and output transformation and B,D are weights for input transformations [6].

Kalman Filters The Kalman filter gained large popularity in fields such as engineering, statistics, and neuroscience and filters data points with respect to a probabilistic latent state and their expected precision. Typically, dynamics and observation models are linear and the observed noise and the latent states are modeled as Gaussian distributions. Similar to the previously discussed model classes, the Kalman filter can be described in SSM form:

$$\begin{aligned} z[t+1] &= Az[t] + Bu[t] + v \\ y[t+1] &= Cz[t+1] + w[t] \end{aligned} \tag{7}$$

with hidden states h_t at discrete timesteps t. Correspondingly to the deterministic IIR filter, the weights of the transition matrix A describe the linear dynamics. The weights of matrix B and C parameterize the observation model. Weights B transform the control inputs u, i.e. known inputs to the system and C map from inferred state to the sensory prediction. Finally, v and w are white noise Gaussian processes with mean zero. The Gaussian prior $p(z_{t+1})$ and posterior distribution $p(z_{t+1} \,|\, y_{1\ldots t}, x_t)$ of the Kalman filter are parameterized by their sufficient statistics, the mean μ and covariance matrix Σ_z [10, 15].

Gradient-Based Predictive Coding Gradient-based predictive coding, as described in has been applied to an approximation of the exact inference in the Kalman filter [10]. In the simplest case, without observations or control inputs, we have a state space model of the form

$$\begin{aligned} z[t+1] &= Az[t] \\ y[t+1] &= Hz[t+1] \end{aligned} \tag{8}$$

where A and H are learnable matrices for the state transition dynamics and the observation model respectively.

Following [10], we define the loss function of the predictive coding filter as:

$$argmin_{\mu_{t+1}} L = argmax_{\mu_{t+1}} p(y_{t+1} \mid z_{t+1}) p(z_{t+1} \mid z_t) \tag{9}$$

In this formulation, weights A and H and the inferred hidden state z (or, more specifically, its mean μ and variance ϵ_z parameters) can be updated using gradient descend based on the precision weighted prediction errors local to the layer [10]:

$$\frac{dL}{d\mu_{t+1}} = -H^T \Sigma_z \epsilon_z + \Sigma_x \epsilon_x, \quad \frac{dL}{dA} = -\Sigma_x \epsilon_x \mu_t^T, \quad \frac{dL}{dC} = -\epsilon_y \mu_{t+1}^T \tag{10}$$

with sensory prediction errors $\epsilon_y = y - H\mu_{t+1}$ and state prediction errors $\epsilon_z = \mu_{t+1} - A\mu_t$ [10]. Intuitively speaking, this means that each layer optimizes the quality of its signal predictions $p_{y_{t+1}} = H\mu_{t+1}$ and of its state predictions $p_{\mu_{t+1}} = A\mu_t$. As this optimization process happens locally informed and in parallel for each optimized variable, many different possible outcomes decrease the prediction error. E.g., quickly adapting observation weights H induce different latent states than a slowly optimized observation model. Similarly, missing accuracy in the observation model might be compensated by hidden state optimization.

A more general form of the predictive coding SSM includes additional weights for control inputs u and observed inputs x:

$$\begin{aligned} z[t+1] &= Az[t] + Bu[t] \\ y[t+1] &= Hz[t+1] + Dx[t] \end{aligned} \tag{11}$$

In summary, we see that single layer predictive coding models and Kalman filters can be represented using the same SSM as IIRs and RNNs (excluding nonlinearities), but additionally differentiate between control and observed inputs.

3 Hierarchical Predictive Coding of Audio

To create a hierarchy of layers with local computations, we can augment the predictive coding SSM mentioned in equation 11 with two sets of weights, F and G. These weights modulate the influence of the layer's own latent state z in comparison to a top-down prediction of this state z_{td} provided by a higher layer:

$$\begin{aligned} z[t+1] &= FAz[t] + GAz_{td}[t] + Bu[t] \\ y[t+1] &= Hz[t+1] + Dx[t] \end{aligned} \tag{12}$$

and denote the weighted state prediction from current and next higher layer with $\hat{z} = Fz + Gz_{td}$. In all experiments, we ignore control inputs u, which could receive known

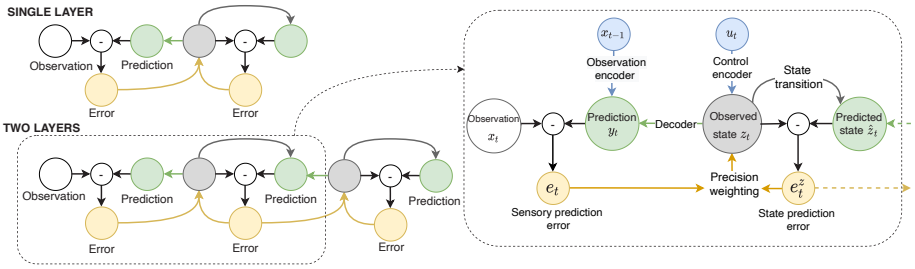


Fig. 2. Predictive Coding network for hierarchical Kalman filtering: At each timestep t, predictions y_t are generated from a latent state z_t using decoder weights that are optimized towards the sensory prediction error e_t between observation x and prediction y. Future latent states z_{t+1} are computed with learnable transition weights. The transition weights are optimized towards the state prediction error e_t^z between predicted state $\hat{z}_t$ and the next inferred state z_t. Hidden PC layers minimize the prediction error e_t^z from a "top-down" prediction of the state. The hidden state z is optimized towards sensory and state prediction error e_t and e_t^z and creates a balance between outgoing and incoming predictions. Optional encoders allow to predict with respect to past observations x_{t-1} or control inputs u.

additional (action) signals and feed past observations x_{t-1} to the observation encoder for the filtering task presented in section 4.3.

The state prediction error now includes the additional input and weights:

$$\epsilon_z = \mu[t+1] - FA\mu[t] - GA\mu_{td}[t] \tag{13}$$

Figure 2 shows an overview of a single layer predictive coding model and how multiple layers can be connected through locally informed predictions and prediction error signals. More precisely speaking, the lowest PC layer directly predicts audio inputs and receives prediction error e_t at every timestep. In contrast, hidden PC layers predict the hidden latent states ("cause units") of the lower layer and receive state prediction error e_t^z. Both lowest and hidden PC layers additionally optimize the weights of their transition model that maps from currently inferred state z_t to the next state z_t+1. We can interpret weights F and G as part of the prediction units that produce the optimal state predictions z_{t+1} given the transition model A. Finally, the latent state z_t+1 is optimized in parallel via gradient descent to minimize the summed precision weighted prediction error $e_t+e_t^z$ local to the respective layer.

We use an overlap-and-add processing approach which is commonly used in DSP, meaning that the PCN processes audio signals in overlapping sequences. For all experiments, the lowest PCN layer processes these sequences sample-by-sample. Hidden layers have identical update frequencies. We found that sequence sizes between 16 and 2048 frames provide meaningful results. The hop-length was set to half the sequence length.

3.1 Audio Analysis and Synthesis with Predictive Coding

Assuming purely linear prediction and a well-trained model, using the PCN for audio resynthesis is possible by reverting the process that computes the residual signal at timestep t (i.e. linear prediction error) from the prediction during analysis. Figure 1 b) shows an overview of the steps for synthesis and analysis given at the lowest layer of a hierarchical predictive

coding model. While this is not the only possible approach to analyze and synthesize signals with predictive coding networks, it has the advantage of relatively exactly replicating the approach taken in LPC. In LPC the coefficients minimizing the squared error during the linear prediction of the next sample resemble compressed versions of the resonances (typically formants in speech coding) and allow the signal to be transmitted with high compression rates through block-wise filter coefficients and down-sampled residual signals. For linear prediction, this LPC residual signal is equal to the prediction error that arises in (gradient-based) predictive coding.

Assuming linear PCN weights and audio with stationary parts, we expect that resonant parts of the audio are gradually removed from the residual. Added hierarchy and non-linear activations will affect the meaning of the first layer's residual signal, e.g. through emerging attentional processes.

4 Results

4.1 Beat Tracking

In order to quantitatively assess the possibility to extract music information from raw audio using prediction errors, we resort to a beat tracking task using two datasets: The SMC MIREX dataset is commonly used for beat tracking evaluation [16]. Our second evaluation is based on finger tapping recordings in the NMED-T dataset that focuses on electroencephalographic (EEG) recordings during music perception [17]. We choose an approach similar to the predominant local pulse (PLP) method described in Grosche et al. [18] and predict beat timings based on a local enhancement of a novelty function. The novelty function in [18] is based on spectral flux, the spectral difference between subsequent Fourier transformed audio inputs. We feed Fourier transformed audio inputs to the PCN (this being the only place where the PCN inputs are not audio samples) and use the prediction error from a single layer PCN to compute the novelty curve. Wherever possible, we use the same FFT parameters as used in Grosche et al. [18] but do not tune any other hyper parameters. For comparison to other approaches, we report the F-measure and two continuity-based metrics: CMLt, measuring correctly tracked beats at the metrical level, and AMLt, which allows variations such as double, half or offbeat variations [19]. All evaluations are based on the mir_eval package [20]. Next to the PLP model, we compare our approach to established baselines: A dynamic Bayesian network from [21] and the dynamic programming approach from [22]. Table 1 shows resulting scores on both datasets.

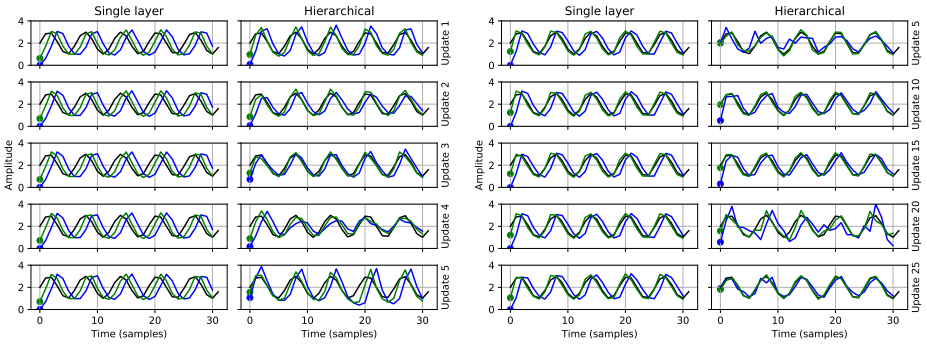
Table 1. Beat tracking evaluation.

SMC MIREX	F-Score	CMLt	AMLt
Ellis [22]	0.339	0.162	0.315
Grosche [18]	0.360	0.071	0.221
Böck - online [21]	**0.521**	**0.363**	**0.433**
PCN (ours)	0.205	0.108	0.201

NMED-T	F-Score	CMLt	AMLt
Ellis [22]	0.277	**0.195**	**0.473**
Grosche [18]	0.305	0.037	0.125
Böck - online [21]	0.092	0.105	0.280
PCN (ours)	**0.321**	0.111	0.295

Interestingly, with respect to the F-Measure, our method outperforms the baselines on the NMED-T dataset but delivers the worst performance on the SMC dataset. This indicates a useful performance on genres with salient rhythmical features, as the NMED-T dataset was designed focusing on Pop songs with clear rhythms. The SMC dataset features many songs with soft onsets, such as strings, where the novelty function from the prediction error is not sufficient. We hope that these encouraging results motivate future work with improved tracking based on predictive coding.

4.2 Audio Filtering with Top-Down Predictions



a) Repeated audio prediction with 10 state updates per timestep and 5 updates of the sequence prior.

b) Repeated audio prediction with 15 state updates per timestep and 25 updates of the sequence prior.

Fig. 3. a) Repeated prediction of a constant sine wave with single layer (left) and hierarchical PCN with two layers (right). The hierarchical model learns a top-down state prior for the sequence, while the single layer model has only local context. When convergence in the lowest layer is not guaranteed, such as with too few gradient descent steps or with inappropriate initialisation of precision, only the hierarchical model correctly tracks the incoming signal. b) With increased gradient steps for state inference in the lowest layer both single-layer and hierarchical PCN eventually show accurate posterior predictions (green). Predictions from the state prior (blue) improve only for the hierarchical model.

Figure 3 shows examples for repeated block-wise prediction of the same audio input with a single layer PCN and a hierarchical PCN with two layers for different gradient steps. In both networks, the inferred state and transition weights of the lowest layer are reset after each sequence prediction. This means that predictions in the single layer PCN are based on local information, i.e. the previously seen samples in the sequence. The hierarchical PCN keeps a top-down prediction of the lower layer's hidden state, providing refined contextual information for each prediction. This learnable state prior noticeably leads to a shifted starting point for the lowest layer in the hierarchical PCN in Fig. 3 a), where the lowest layer has not enough time to converge properly. When initialised with optimised parameters, both variants are able to approximate the target audio to a reasonable degree and the differences in prediction (and associated prediction errors) are largely restricted to the start of the sequence, as visible in Fig. 3 b). This indicates that minimizing prediction error can be solved through

online inference in independent trials as well as through the more gradual process of weights learning when information between trials is carried over. As noticeable in both Fig. 3 a) and b), the learning dynamic of the hierarchical model is significantly more dynamic, since the weighting of the top-down state prior is slightly adapted at each timestep.

The posterior predictions, indicated in Fig. 3 with green lines, show that the lowest PCN layer does not directly adapt to the top-down prior, but needs some time to tune the remaining weights to this additional source of information. When the top-down prior is correctly integrated, however, the hierarchical model quickly improves over the single layer model, especially with parameter initialisation that prevents full convergence of prediction errors in the lowest layer.

4.3 Replicating Filter Transfer Functions

We tested the possibility to simulate a Butterworth low-pass (LP) filter, which is widely in various DSP applications. Figure 4 shows input and output audio signals to the targeted LP filter and the corresponding in and outputs of a PCN. We test PCNs with single and two layers on a constantly ascending sine wave tone superimposed on constant white noise. Both PCN variants are able to replicate the desired transfer function of the LP filter and show the desired high frequency content removal.

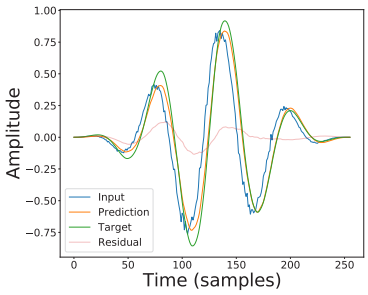


Fig. 4. Replicating an order 2 Butterworth LP filter. LP filter and PCN remove high frequency contents and have comparable output magnitudes. As the prediction starts with randomized states and without top-down prior, the prediction error (red) is higher at the sequence start.

5 Conclusion

We presented a gradient-based predictive coding model for audio analysis and synthesis. The hierarchical model targets biological plausibility through locally informed updates while still being efficient and accurate enough to replicate classical DSP tasks like filtering and beat tracking. We reviewed the similarities between the autoregressive state-space models underlying predictive coding, IIR filters, recurrent neural networks, and Kalman filtering. The model provides a basis for future work that could approach more complex DSP applications or subjectivity in artificial music perception.

References

1. Kumar, S., Sedley, W., Nourski, K. V., Kawasaki, H., Oya, H., Patterson, R. D., Howard III, M. A., Friston, K. J., Griffiths, T. D.: Predictive coding and pitch processing in the auditory cortex. Journal of Cognitive Neuroscience, 23(10):3084–3094 (2011)
2. Friston, K., Kiebel, S.: Predictive coding under the free-energy principle. Philosophical Transactions of the Royal Society of London B: Biological Sciences, 364(1521):1211–1221 (2009)
3. Millidge, B., Tschantz, A., Buckley, C.: Predictive coding approximates backprop along arbitrary computation graphs. arXiv preprint arXiv:2006.04182 (2020)
4. Skerritt-Davis, B., Elhilali, M.: Computational framework for investigating predictive processing in auditory perception. Journal of Neuroscience Methods, 109177 (2021)
5. Miguel, M. A., Sigman, M., Fernandez Slezak, D.: From beat tracking to beat expectation: Cognitive-based beat tracking for capturing pulse clarity through time. PloS one, 15(11):e0242207 (2020)
6. Kuznetsov, B., Parker, J. D., Esqueda, F.: Differentiable IIR filters for machine learning applications. In: Proc. Int. Conf. Digital Audio Effects (eDAFx-20), pp. 297–303 (2020)
7. Adams, R. A., Friston, K. J., Bastos, A. M.: Active inference, predictive coding , cortical architecture. In: Recent Advances on the Modular Organization of the Cortex, 97–121. Springer (2015)
8. Marino, J.: Predictive coding, variational autoencoders, and biological connections. arXiv preprint arXiv:2011.07464 (2020)
9. Friston, K.: Hierarchical models in the brain. PLoS computational biology, 4(11), e1000211 (2008)
10. Millidge, B., Tschantz, A., Seth, A., Buckley, C.: Neural kalman filtering. arXiv preprint arXiv:2102.10021 (2021)
11. Irshad A., Salman M.: State-space approach to linear predictive coding of speech—a comparative assessment. In: IEEE 8th Conf. on Ind. Electronics and Applications (ICIEA), pp. 886–890. (2013)
12. Kondoz, A. M.: Digital speech: coding for low bit rate communication systems. John Wiley & Sons (2005)
13. O'Shaughnessy, D.: Linear predictive coding. IEEE potentials, 7(1):29–32 (1988)
14. Elman, J. L.: Finding structure in time. Cognitive science, 14(2):179–211 (1990)
15. Kalman, R. E.: A new approach to linear filtering and prediction problems. (1960)
16. Holzapfel, A., Davies, M. E., Zapata, J. R., Oliveira, J. L., Gouyon, F.: Selective sampling for beat tracking evaluation. IEEE Transactions on Audio, Speech, and Language Processing, 20(9):2539–2548 (2012)
17. Losorelli, S., Nguyen, D. T., Dmochowski, J. P., Kaneshiro, B.: NMED-T: A tempo-focused dataset of cortical and behavioral responses to naturalistic music. In: Proc. of the 18th Int. Society for Music Information Retrieval Conference (2017).
18. Grosche P., Muller, M.: Extracting predominant local pulse information from music recordings. IEEE Transactions on Audio, Speech, and Language Processing, 19(6):1688–1701 (2010)
19. Davies, M. E., Degara, N., Plumbley, M. D.: Evaluation methods for musical audio beat tracking algorithms. Queen Mary University of London, Centre for Digital Music, Tech. Rep. C4DM-TR-09-06, (2009)
20. Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., Ellis, D.: mir_eval: A transparent implementation of common mir metrics. In: Proc. of the 15th Int. Society for Music Information Retrieval Conference (2014)
21. Böck, S., Krebs, F., Widmer, G.: A multi-model approach to beat tracking considering heterogeneous music styles. In: Proc. of the 15th Int. Society for Music Information Retrieval Conference, (2014)
22. Ellis, D. P.: Beat tracking by dynamic programming. Journal of New Music Research, 36(1):51–60 (2007)
23. Schwartenbeck, P., Passecker, J., Hauser, T. U., FitzGerald, T. H., Kronbichler, M., Friston K. J.: Computational mechanisms of curiosity and goal-directed exploration. Elife, 8:e41703 (2019)

Zero-shot Singing Technique Conversion

Brendan O'Connor[1], Simon Dixon[1], and George Fazekas[1] ⋆

Centre for Digital Music, Queen Mary University of London, UK
b.d.oconnor@qmul.ac.uk

Abstract. In this paper we propose modifications to the neural network framework, AutoVC [17] for the task of singing technique conversion. This includes utilising a pretrained singing technique encoder which extracts technique information, upon which a decoder is conditioned during training. By swapping out a source singer's technique information for that of the target's during conversion, the input spectrogram is reconstructed with the target's technique. We document the beneficial effects of omitting the latent loss, the importance of sequential training, and our process for fine-tuning the bottleneck. We also conducted a listening study where participants rate the specificity of technique-converted voices as well as their naturalness. From this we are able to conclude how effective the technique conversions are and how different conditions affect them, while assessing the model's ability to reconstruct its input data.

Keywords: Voice synthesis, singing synthesis, style transfer, neural network, singing technique, timbre conversion, conditional autoencoder, sequential training, latent loss

1 Introduction

Voice conversion (VC) is the task of converting the timbre of the voice so that the linguistic content is perceived to be spoken by a different person. It has been explored in relation to both singing and speech, which both possess different attributes consideration. Singing voice analysis is considerably more focused on sustained notes, harmonic/rhythmic structure, and relative pitch. In speech, these musical values are non-existent. Instead there is greater emphasis on aperiodic aspects, such as consonant utterances and rapidly shifting spectral envelopes. Tasks like VC and text-to-speech are in far more demand in the industry than singing-related tasks, and have therefore monopolised the spotlight in voice analysis and synthesis research. The latest approaches towards VC achieving state-of-the-art conversions utilise probabilistic machine learning techniques. Public domain speech datasets also vastly overshadow singing datasets in size and availability [11], and so there is still much to be explored in relation to singing analysis and synthesis.

In this paper we tackle the task of singing technique conversion (STC) - the task of converting a singing technique without affecting the perceived identity of the singer, musical structure or linguistic content. We define singing technique as the method of

⋆ This research is funded by the EPSRC and AHRC Centre for Doctoral Training in Media and Arts Technology (EP/L01632X/1).

voice production to achieve different timbres by adjusting the airflow, vocal folds, vocal tract shape, and sympathetic vibrations in the body [5]. We regard STC as a variation of voice conversion (VC), where the possibilities of voice transformation are restricted to be within a realistic variance of timbre for any given singer. We chose the term singing *techniques* as opposed to singing style, due to the latter term's inconsistent use in literature, often referring to a range of very different audio and musical attributes due to its lack of reference to a concrete audio or singing concept.

To achieve STC, we apply a neural network model in the form of the conditioned autoencoder, AutoVC [17]. We discuss certain adaptations made to the architecture and investigate the effects of training it on different permutations of several datasets. To evaluate the model's ability to perform STC, we had participants rate the naturalness of the voice and guess what the target singing technique was supposed to be. Examples of audio used in this listening test can be found online.[1]

Real-time pitch correction algorithms have become commonplace in the music industry and influence the characteristics of modern pop singers today. We believe that the refined task of STC could have a similar influence on music production as it opens up the possibility of artistically manipulating a singer's *performance*, rather than just quantising their pitch. Over the last 5 years, many machine-learning approaches have been proposed to tackle voice transformation for speech (as discussed in the next section), but much less attention has been given to transforming the expression of the singing voice.

2 Related Work

Recent research in VC has been based on neural networks, which have influenced the frameworks proposed in this paper. [15] conditioned an autoencoder (trained on linguistic data) on speaker embeddings generated from a separately trained classifier network. During inference, these embeddings could be replaced to achieve VC. AutoVC [17] adapted this method to work with spectrograms, which will be described in detail in Section 4. This was improved upon by conditioning the network on pitch contours to enforce prosody during conversion [18], and further disentanglement was achieved for timbre, pitch contours, rhythm and utterances simultaneously by utilising 3 separate bottlenecks with different restrictions [19]. [26] achieve VC by using vector quantisation to separate speaker and content information, and later utilised U-nets [21] to compensate for information lost during vector quantisation.

The application of the variational autoencoder (VAE) is well suited for 'many-to-many' conversions (where all examples used for inference are seen during training). [16] use fully convolutional VAEs, conditioned on acoustic features, to perform VC. They combine spectral features of both converted and unconverted reconstructed audio in order to avoid over-smoothing - a known issue with VAEs. While VAEs present an elegant framework, they produce 'blurry' results. Generative Adversarial Networks (GANs) have been known to reproduce better quality reconstructions of images than VAEs. However as they come without an autoencoder they are harder to train and suffer from 'mode collapse', and there has not yet been an elegant proposal for combining

[1] https://github.com/Trebolium/singing_technique_conversion

VAEs with GANs [22]. The use of VAEs has the added benefit of utilising unsupervised learning, which bypasses the issue of low resources regarding labelled singing datasets. [6] used Gaussian-Mixture VAEs (GMVAEs) for controllable speech synthesis, modelling the different attributes of speech as separate prior distributions before combining them in a VAE. For singing voice conversion, [13] adapted AutoVC by conditioning the network on pitch contours transposed to a suitable register for the converted singing, achievable through the implementation of a vocoder. [8] utilised a Wasserstein-GAN framework, using a decoder for pitch contours and another for generating 'formant masks'. The product of these two decoders is the estimated mel-spectrogram for singing. They later explored the capabilities of this framework to achieve timbre and singing style disentanglement [9], where a *singing query* is converted into a singer identity embedding and used to condition both the pitch skeleton and formant-mask encoders on pitch modulation style and singer timbre, respectively. [10] present the only other research we know of that addresses STC. They use GMVAEs to model singer and technique information to perform many-to-many conversions using a VAE architecture that utilises a convolutional recurrent neural network (CRNN) architecture.

The issue remains however, of what can be done with singing datasets which are small and few. [13] notes that the generalisation of the AutoVC framework allows it to be utilised as a Universal Background Model. [1] synthesise monophonic singing datasets by superimposing pitch contours on existing speech datasets. [3] use several autoencoder instances, trained separately on vocoder spectral data and music mixtures, while being conditioned on shared content embeddings and 1-hot speaker embeddings to produce a final network that is singer-independent and generates monophonic singing from musical mixtures. [12] generate novel speaker embeddings by combining embeddings from existing singers as a method of data augmentation.

3 Architecture

We use the AutoVC framework [17] for singing technique transformation, due to its elegant method of applying disentanglement. It is also capable of converting between source and target examples that have not been seen in the training datasets (zero-shot conversion). In AutoVC, a standard autoencoder architecture is conditioned on speaker embeddings that uniquely describe the timbre of a speaker to perform VC on spectrograms. These embeddings are generated by a pretrained speaker verification network [24]. The spectrograms are concatenated with these speaker embeddings, and fed through an encoder E_c, after which the encoded information is again concatenated with speaker embeddings before being fed to the decoder D_c. This conditioning, combined with careful calibration of an appropriate bottleneck size, allows the autoencoder to disentangle speaker timbre from utterance information. AutoVC also contains a 'postnet' convolutional layer which is appended to the decoder to further develop a refined spectrogram from the decoder's output. After training, the speaker embeddings concatenated at the bottleneck can be swapped out to achieve VC. The loss function for AutoVC is a weighted combination of the self-reconstruction loss for both the decoder ($L_{decoder}$) and the postnet ($L_{postnet}$) output spectrograms, and the latent loss (L_{latent}). The latent loss represents the difference between the bottleneck's embedding $E_c(x)$ for the input x

and its reconstructed form $E_c(\hat{x})$. This is summarised in Equation 1, where μ and λ are empirically determined weights. Further details of AutoVC's architecture are given by [17], which we follow in our implementation except for several adjustments discussed in this section.

$$L_{total} = L_{decoder} + \mu L_{postnet} + \lambda L_{latent}. \quad (1)$$

We will herein refer to our implementation of AutoVC as AutoSTC to reflect its purpose of STC. To facilitate this, we developed our own singing technique encoder (STE) to replace the external speaker encoder that was used in the original implementation. The STE is initially trained as a classifier. It takes a mel spectrogram as input, which is split into chunks of 0.5 seconds. These are fed in parallel through a neural network consisting of four 2D-convolutional layers (each of which is followed by batch normalisation, ReLU activation and max-pooling), two dense layers, two BLSTMs, a simplified attention mechanism [20], two more dense layers and finally a classification layer. This architecture was adapted from the VAE used by [10] and influenced by [4]. This network is able to achieve 86% accuracy when classifying singing techniques within a test set of VocalSet (detailed in Section 4, while our implementation of a 1D convolutional network on the waveform data as described by [25] only scored 57%. During conversion, the STE's embedding preceding the classification layer are used for concatenation and conditioning with AutoVC as described above in place of the external speaker encoder embeddings.

4 Training and Inference

The Vocalset dataset [25] used to train the STE consists of recordings of 20 singers performing several musical exercises with different singing techniques. We chose a subset containing the techniques *belt, straight, vibrato, lip trill, vocal fry* and *breathy*, trimming off excess files that appear in one class but not the other, to yield a balanced class subset of 1182 examples (roughly 8K seconds). As the dataset is so small we only partition it into training and test sets by 8:2.

As [13] showed that the sequence of training of different datasets is important, AutoSTC was trained using subsets taken from VocalSet, VCTK [23] and the raw singer recordings from MedleyDB [2] in various permutations. All data was sampled at 16kHz and transformed into 80-bin mel spectrograms. While being trained on one dataset, AutoSTC was simultaneously tested on test sets from all three datasets in between training iterations (the VocalSet test set was the same set omitted when training the STE). We recorded the number iterations and loss values for each dataset where the loss showed no further improvement into Table 1, and transferred the saved neural network parameters of a nearby checkpoint to the proceeding dataset training session in the sequence. We trained AutoSTC once for every permutation of the datasets. Table 1 shows that the order in which datasets are fed to the network does have a considerable impact on its loss. The paths `Vc->Vs->Md` (spanning 750k training steps) and `Vc->Md->Vs` (500k steps) led to the lowest loss values for MedleyDb and VocalSet reconstruction respectively, and were used to train models that generated the examples used in our listening test (see Section 5).

Table 1. Shows losses and training iterations (in parentheses) for VocalSet (left) and MedleyDB (right) along alternative paths. The optimum training path is highlighted in bold from left to right. Training that leads to an increase in loss is indicated with a circumflex, at which point that path is abandoned. For space, the dataset names are shortened as follows: VCTK:Vc, VocalSet:Vs, MedleyDB:Md.

Loss-Iteration for Vs					
Vc	**0.0653(300k)**	Vs	0.0274(100k)	Md	^
		Md	**0.0386(150k)**	**Vs**	**0.0268(50k)**
Vs	0.0347(150k)	Vc	^	Md	-
		Md	^	Vs	-
Md	0.0500(200k)	Vs	0.0290(50k)	Vc	^
		Vc	^	Vc	-

Loss-Iteration for Md					
Vc	**0.0479(500k)**	**Vs**	**0.0474(150k)**	**Md**	**0.0265(100k)**
		Md	0.0295(150k)	Vs	^
Vs	0.0562(150k)	Vc	0.0474(100k)	Md	0.0301(50k)
		Md	0.0370(100k)	Vs	^
Md	0.0367(150k)	Vs	^	Vc	-
		Vc	^	Vc	-

We found the L1 loss between decoder/postnet self-reconstructions and the input encouraged better convergence over L2 loss. We also tested the impact of excluding the latent loss for 100K steps for both VC and STC tasks. Results showed that training without latent loss performs significantly better for both tasks. The loss for STC with latent loss was 0.0237 (and 0.0185 without loss), while spectrograms were blurry and the audio lacked microtonal variation or vibrato, leaving a 'bubbliness' artefact in its absence. Vowels were also not consistently reproduced. These shortcomings however are less noticeable for speech than singing. This result is worth highlighting, as latent loss has been used consistently in frameworks of a similar nature [13, 15, 17].

Further preliminary trials allowed us to fine-tune AutoSTC's bottleneck. We analysed the resynthesised audio and noted that the net size of the feature space was more indicative of audio quality than focusing on time/frequency axis fine-tuning separately. We estimated the threshold to be a downsampling factor of 16 for the time-axis, with each timestep containing 32 features. Lower dimensionality representations resulted in deterioration in the reconstructed audio in a very similar manner to that of the network with latent loss included.

5 Experiment Design

To evaluate our proposed network's ability to perform STC, we conducted a listening study, where 19 participants evaluated the converted audio for specificity and naturalness under the different conditions of models, gender, and source/target techniques used. The first of the models (Vs1) was trained on VocalSet alone and converted Vocalset data. The second (Vs2) was trained using the optimum path for Vocalset presented in Table 1 to also convert VocalSet data, while the third (M1) used the optimum path for MedleyDB to convert MedleyDB data. Converted spectrograms were resynthesised using a pretrained wavenet model provided by the author of the AutoVC paper[2]. Each model produced 8 random examples per participant, while adhering to a balanced representation of both gender and subset (train/test) conditions. To evaluate naturalness, we asked participants to consider how synthetic/realistic the voice itself sounded, and rate them on a scale of 1 (very unnatural) to 5 (very natural). In a separate task to evaluate

[2] https://github.com/auspicious3000/autovc

specificity, participants were given a reference recording featuring a converted singing technique along with 6 unlabelled candidate recordings from the same singer to choose from. These candidate recordings were randomly selected from the relevant dataset partition, so that they each featured 1 of the 6 potential target techniques assigned to the reference recording. Participants were asked to select one recording they thought featured a singing technique closest to that of the reference recording, or more if the answer seemed ambiguous. In the case where reference recordings were converted MedleyDB examples, no ground truth labels existed, and so a singer of the same gender was randomly chosen from Vocalset to represent the 6 candidate singing techniques instead. Each of these tasks was presented 24 times. 6 resynthesised recordings of unconverted audio were also evaluated for naturalness. The interface was built using the Web Audio Evaluation Tool [7].

6 Results

The Mean Opinion Score (MOS) for unconverted data was 3.75 ± 0.34, and is important to consider when analysing the results of the study. This highlights the fact that a considerable amount of perceived naturalness has already been lost during the wavenet resynthesis process, and that the MOS values for technique conversion should be considered with this in mind. It is the comparison between conditions that we are interested in.

To calculate the similarity score S for each condition, we used the formula in Equation 2, where P_n is a binary vector reflecting a participant's true/false predictions (identifying whether each candidate technique was the same as what was presented in the reference audio) for the nth task, C_n is a 1-hot vector reflecting the correct technique for the task, and N is the total number of tasks in the given condition. The similarity score is an average count of correct predictions weighted by the reciprocal of the number of predictions made for the corresponding task.

$$S = \frac{1}{N} \sum_{n=1}^{N} \frac{P_n . C_n}{||P_n||_1}. \tag{2}$$

Figure 1 displays the results obtained from the listening study. The top graph displays MOS values for naturalness, with whiskers indicating the confidence intervals. The lower graph displays similarity scores. The combination of these two graphs give us insight into how each of our models perform, and what conditions influence the naturalness and specificity of the converted singing.

We detected from a Spearman's rank analysis that MOS and similarity scores were not significantly correlated. Similarity scores across all conditions measure higher than the chance level (0.16), which suggests that our models have some success in converting to the target techniques. The condition of source-technique groups does not significantly influence recognisability of the converted singing technique. However, the data would suggest that the features of the target techniques *trill* and *breathy* are significantly more distinguishable than the rest. *Vibrato* scored the lowest for similarity, suggesting that this was a particularly difficult technique to synthesise convincingly. The reason for

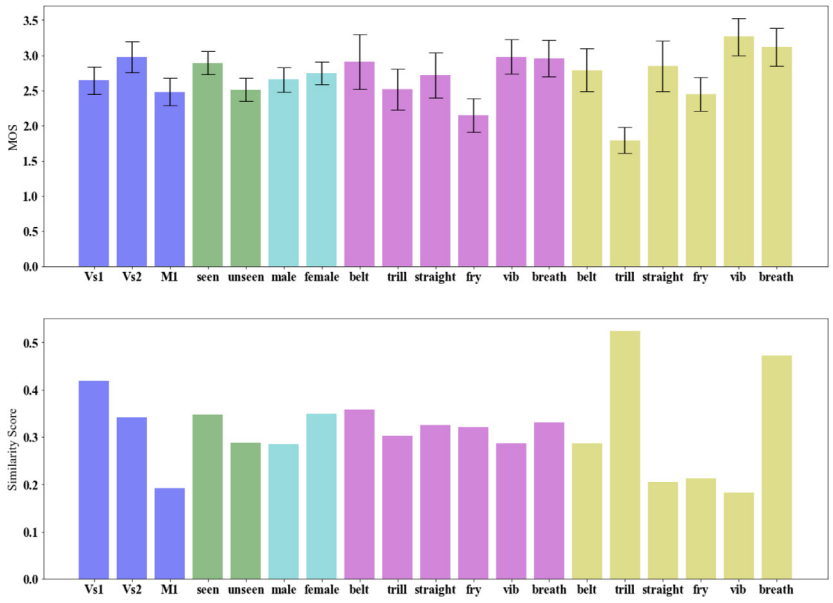


Fig. 1. Top: Bar graph showing naturalness (MOS values and confidence intervals) for all conditions. The colours group together the conditions for (left to right): models, subsets, genders, source technique and target technique. **Bottom:** Bar graph showing similarity scores determined by Equation 2, relative to correct answers.

this is most likely due to the fact that VocalSet, upon which the STE network was trained, contains numerous examples labelled as *belt* and *straight*, while still featuring a considerable amount of frequency modulation (a unique feature of vibrato), making it difficult for AutoSTC to disentangle vibrato from other techniques effectively. It may also be the case that AutoSTC has difficulty disentangling vibrato from pitch contours. Alternatively it is possible that our models instead focused on altering the phonation modes associated with vibrato, which would be considerably less obvious to listeners than identifying whether frequency modulation is occurring.

The inclusion of all datasets in training Vs2 seemingly diminished its ability to accurately convert techniques (although the difference was not statistically significant). The M1 model scored significantly worse than the other models, which tells us that the features learned to generate technique embeddings from the STE network were not generalisable to data outside the dataset the VTE was trained on. There was also no statistically significant difference between gender and subset similarity scores.

In regards to MOS results, the target technique *trill* scored lowest, suggesting that conversions to a trill technique may sound unnatural. Vs2 samples were significantly higher than Vs1 and M1, which suggests that providing the network with multiple datasets does improve its ability to synthesise natural sounding data. The target technique condition *vibrato* scored the highest, but as mentioned above, this may be because the network is making changes more subtle than the frequency modulation which

lessens the amount of transformation required, causing less synthetic artefacts. It is also perfectly possible that participants simply perceive the singing voice to be more natural when vibrato is present.

7 Conclusion

In this paper we have presented a network for vocal technique classification, and the first network to perform zero-shot conversion on singing techniques, achieving above chance level for all tested conditions. We have demonstrated that omitting latent loss and choosing the order in which AutoSTC was fed different datasets significantly diminished its reconstruction loss, improving its ability to reconstruct mel spectrograms. However we can conclude from the results of the listening study that this does not have any significant effect on AutoSTC's ability to perform technique conversion and may even diminish it. We therefore conclude that the features generated by supervised learning on the labelled VocalSet dataset are not sufficient to generalise to recordings of other singers. We also consider that the appearance of frequency modulation in other techniques in VocalSet may have forced the network to give less importance to this vibrato feature (we have however witnessed conversions where frequency modulation was synthesised, but in very limited cases, so we can not rule out the possibility that the AutoSTC framework is incapable of converting singing technique features beyond their spectral filter properties). The findings of our listening study are in agreement the vocal timbre maps generated in our previous research [14].

Augmentation techniques such as those discussed in Section 2 may improve the generalisation of the VTE to unseen data. We would also like to apply the Generalised End-to-End Loss techniques from [24] to the VTE and fine-tune its output embedding size. Due to shortcomings in labelled datasets, we will explore unsupervised/semi-supervised networks such as VAEs. It may also be worth investigating how AutoSTC performs when we condition it on further attributes such as speaker identity, pitch contours and vowel sounds. As we consider STC to be a restricted variation of VC and the fact that there are considerably larger datasets for speech, it may also be worth exploring the effects of pre-training an AutoVC framework for VC before switching its speaker encoder for the singing technique encoder and training it for STC. In future work we will also consider alternative options to the speech-trained wavenet vocoder as this has introduced artefacts to the audio that likely lowered MOS ratings for all audio. We have also observed that AutoSTC was unintentionally able to remove vibrato from singing when underfitting, which may be a capability worth fine-tuning in future work.

References

1. Basak, S., Agarwal, S., Ganapathy, S., Takahashi, N.: End-to-end Lyrics Recognition with Voice to Singing Style Transfer. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 266–270 (2021) https://doi.org/10.1109/ICASSP39728.2021.9415096
2. Bittner, R., Salamon, J., Tierney, M., Mauch, M., Cannam, C., Bello, J.: MedleyDB: A Multitrack Dataset for Annotation-intensive MIR Research. In: 15th International Society for Music Information Retrieval Conference, pp. 155–160. (2014)

3. Chandna, P., Blaauw, M., Bonada, J., Gomez, E: Content Based Singing Voice Extraction from a Musical Mixture. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 781-785 (2020)
4. Choi, K., Fazekas, G., Sandler, M., Cho, K.: Convolutional Recurrentneural Networks for Music Classification. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2392–2396. (2017) https://doi.org/10.1109/ICASSP.2017.7952585
5. Heidemann, K.: A System for Describing Vocal Timbre in Popular Pong. In: Music Theory Online, vol. 22, (2016) https://doi.org/10.30535/mto.22.1.2
6. Hsu, W.-N., Zhang, Y., Weiss, R. J., Zen, H., Wu, Y., Wang, Y., Pang, R.: Hierarchical Generative Modeling for Controllable Speech Synthesis. In: The International Conference on Learning Representations, p. 27. New Orleans, LA, USA (2019)
7. Jillings, N., Moffat, D., De Man, B., Reiss, J. D.: Web Audio Evaluation Tool: A browse-based listening environment. In: 12th Sound and Music Computing Conference. Maynooth, Ireland. (2015)
8. Lee, J., Choi, H.-S., Jeon, C.-B., Koo, J., Lee, K.: Adversarially Trained End-to-end Korean Singing Voice Synthesis System. arXic preprint arXiv:1908.01919 (2019)
9. Lee, J., Choi, H.-S., Koo, J., Lee, K.: Disentangling Timbre and Singing Style with Multi-Singer Singing Synthesis System. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7224–7228 (2020) https://doi.org/10.1109/ICASSP40776.2020.9054636
10. Luo, Y.-J., Hsu, C.-C., Agres, K., Herremans, D.: (2019). Singing Voice Conversion with Disentangled Representations of Singer and Vocal Technique Using Variational Autoencoders. arXiv preprint arXiv:1912.02613 (2019)
11. Meseguer-Brocal, G., Cohen-Hadria, A., Peeters, G.: Creating DALI,a Large Dataset of Synchronized Audio, Lyrics, and Notes. In: Transactions of the International Society for Music Information Retrieval, vol. 3 pp. 55-67 (2020) https://doi.org/10.5334/tismir.30
12. Nachmani, E., Wolf, L.: Unsupervised Singing Voice Conversion. arXiv preprint arXiv:1904.06590 (2019)
13. Nercessian, S.: Zero-shot Singing Voice Conversion. In: Proceedings of the International Society for Music Information Retrieval Conference (2020)
14. O'Connor, B., Dixon, S., Fazekas, G.: An Exploratory Study on Perceptual Spaces of the Singing Voice. In: The 2020 Joint AI Conference on Music Creativity, vol. 1, Stockholm, Sweden (2020)
15. Jia, Y., Zhang, Y., Weiss, R. J., Wang, Q., Shen, J., Ren, F., Wu, Y.: Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis. arXiv preprint arXiv:1806.04558 (2019)
16. Kameoka, H., Kaneko, T., Tanaka, K., Hojo, N.: ACVAE-VC: Non-parallel many-to-many voice conversion with auxiliary classifier variational autoencoder. arXiv preprint arXiv:1808.05092 (2020)
17. Qian, K., Zhang, Y., Chang, S., Yang, X., Hasegawa-Johnson, M.: AutoVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss. In: 36th International Conference of Machine Learning, vol. 47, pp. 5210-5219 (2019)
18. Qian, K., Jin, Z., Hasegawa-Johnson, M., Mysore, G. J.: F0-Consistent Many-To-Many Non-Parallel Voice Conversion Via Conditional Autoencoder. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6284–6288 (2020) https://doi.org/10.1109/ICASSP40776.2020.9054734
19. Qian, K., Zhang, Y., Chang, S., Cox, D., Hasegawa-Johnson, M.: Unsupervised speech decomposition via triple information bottleneck. In: 37th International Conference on Machine Learning, vol. 119, pp. 7792–7802 (2020)

20. Raffel, C., Ellis, D. P. W.: Feed-Forward Networks with Attention Can Solve Some Long-Term Memory Problems. arXiv preprint arXiv:1512.08756 (2016)
21. Ronneberger, O., Fischer, P., Brox, T: U-Net: Convolutional Networks for Biomedical Image Segmentation. In: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (eds.) Medical Image Computing and Computer-Assisted Intervention, vol. 9351, pp. 234–241. (2015) https://doi.org/10.1007/978-3-319-24574-428
22. Tolstikhin, I., Bousquet, O., Gelly, S., Schoelkopf, B.: Wasserstein Auto-Encoders. arXiv preprint arXiv:1711.01558 (2019)
23. Veaux, C., Yamagishi, J., MacDonald, K.: CSTR VCTK Corpus: English multi-speaker Corpus for CSTR Voice Cloning Toolkit. (2017) https://doi.org/10.7488/ds/2645
24. Wan, L., Wang, Q., Papir, A., Moreno, I. L.: Generalized End-to-End Loss for Speaker Verification. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4879–4883 (2018) https://doi.org/10.1109/ICASSP.2018.8462665
25. Wilkins, J., Seetharaman, P., Wahl, A., Pardo, B. (2018).: VocalSet: A Singing Voice Dataset. 19th International Society for Music Information Retrieval Conference, Paris ,France pp. 468–474 (2018)
26. Wu, D.Y., Chen, Y.H., Lee, H.Y.: One-Shot Voice Conversion by Vector Quantization. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2020)

Audio-Tactile Perception of Roughness

Madeline Fery[1], Corentin Bernard[1,2], Etienne Thoret[1,3,4], Richard Kronland-Martinet[1], Sølvi Ystad[1]

[1] Aix Marseille Univ, CNRS, PRISM, Marseille, France
[2] Aix Marseille Univ, CNRS, ISM, Marseille, France
[3] Institute of Language Communication and the Brain (ILCB), Marseille, France
[4] Aix Marseille Univ, LIS, CNRS, Marseille, France

etienne.thoret@univ-amu.fr; bernard@prism.cnrs.fr

Abstract. Auditory roughness is a perceptual attribute at the basis of phenomena such as consonance and dissonance in music. The psychophysical correlates of this attribute are often studied by combining two monochromatic tones slightly separated in frequency, leading to more or less rapid beatings. Interestingly, roughness is not limited to the auditory modality and it is possible to evoke the same kind of sensation through the tactile modality by using a vibrotactile actuator. Whether or not audio and tactile modalities share the same perceptual roughness properties is still an open question that may reveal common sensory processes between the two modalities. Here we investigate this question in 2 pairwise comparison experiments unveiling roughness curves in audio and tactile modalities. The results reveal similar roughness curves in both modalities, which suggests a common way of processing and perceiving beatings.

Keywords: Audio-tactile, Roughness, Beatings, Critical bands

1 Introduction

Auditory roughness probes the fundamental ability of audition to disentangle harmonic stimuli, a fundamental skill to perceive speech (Arnal et al., 2015), and music (Helmholtz, 1885; Plomb & Levelt, 1965) properties. This phenomenon can be described as the perception of very fast fluctuations in sounds. To understand how our ears deal with complex mixtures of harmonics (Vassilaki, 2001), a historical body of works has used basic stimuli by combining monochromatic tones. It is now well known that for stimuli composed of two monochromatic tones, the sensation of roughness is driven by the space between the frequencies of the components. The roughness first increases when the frequency ratio between components increases and reaches a maximum before it decreases with respect to the increasing frequency ratio. Figure 1 presents a typical auditory roughness curve for a sum of two monochromatic tones $s(t) = sin(2\pi f_1 t) + sin(2\pi f_2 t)$. When the frequency ratio $\alpha = f_2/f_1$ is small, the combination of tones tends to be perceived as one tone slowly modulated by the other one. When the frequency ratio increases, a sensation of roughness appears. As α

becomes even larger, the perceived roughness falls and two the two tones are perceived separately. This theoretical roughness curve is defined by $r(f_1, f_2) = e^{-b_1 s(f_2 - f_1)} - e^{-b_2 s(f_2 - f_1)}$ with $b_1 = 3.5$, $b_2 = 5.75$, $s = \frac{0.24}{s_1 f_1 + s_2}$, s_1=0.0207 and s_2 = 18.96 (Vassilakis, 2001).
Such a phenomenon reveals the existence of auditory critical bands, a fundamental characteristic of auditory filters (Terhardt, 1974). While the auditory perception of such phenomena has been largely studied, it is not known whether other modalities such as touch, elicit similar behaviors (Makous et al., 1995). Interestingly, it is possible to produce similar stimuli as in auditory experiments with vibrotactile actuators. It would therefore be interesting to check whether roughness perception is shared between auditory and tactile modalities. This would strongly suggest that they might also share mechanistic properties during the processing of vibrations. In a larger multisensory perspective, we may wonder how this information is processed and in particular how the information is shared between the auditory and tactile inputs. Is it possible to influence auditory roughness with tactile feedback? And conversely, might a smooth surface be perceived as rough when touched in presence of a rough sound?

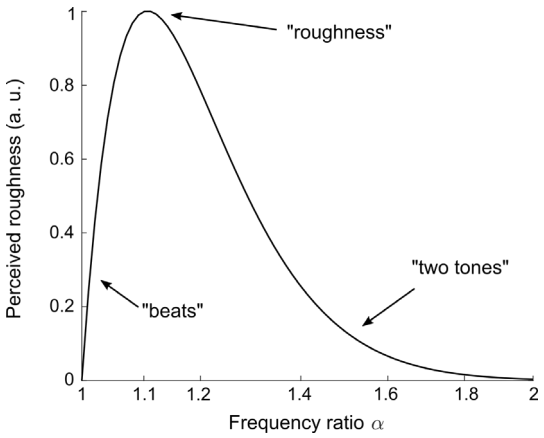


Fig. 1. Typical auditory roughness curve of a sum of two monochromatic tones and description of the three kinds of sensations provoked by pairs of pure tones for f_1=200 Hz (Vassilakis, 2001).

In this paper, we investigated the perception of roughness through an experiment that was divided in two parts, one with audio stimuli and the other one with tactile stimuli. The results are presented as audio and tactile roughness curves obtained from pairwise comparisons.

2 Method

2.1 Participants

18 subjects, 8 women and 10 men (Mean=30 years old, between 21 and 57 years old), 15 right-handed and 3 left-handed, voluntarily took part in the experiment. None of them reported having any auditory problems or skin concerns. The participants gave their informed consent before the experiment. The experiment lasted about 1 hour.

2.2 Audio stimuli

Audio stimuli were pairs of sounds, each composed of the sum of two monochromatic tones of frequencies f_1 and $f_2=\alpha f_1$, of duration 1 second, and separated by 800ms. α is a coefficient between 1 and 2 that determines the frequency ratio between the two frequencies. When α=1, the frequencies are the same ($f_1=f_2$), which corresponds to unison, and when α=2 the tones are separated by an octave $f_2=2f_1$: $s(t) = sin(2\pi f_1 t) + sin(2\pi f_2 t)$.

Twelve values of α were chosen (1, 1.01, 1.02, 1.03, 1.05, 1.10, 1.15, 1.20, 1.25, 1.35, 1.50, 2.00) leading to 66 comparison pairs for one block. Audio stimuli were compared for 6 frequency conditions (f_1 = 50, 100, 200, 300, 600, 1200 Hz) leading to 6 blocks of 66 pairs (=396 pairs). Sounds were presented through Sennheiser HD-650 headphones at a sampling rate of 44100 Hz powered by a Pioneer A-209R audio amplifier.

2.3 Tactile stimuli

Tactile stimuli were generated with the same procedure as the audio stimuli for 4 frequency conditions only (f_1 = 50, 100, 200, 300 Hz). Frequencies above 800 Hz are indeed not perceptible by the human tactile sensory system (Verrillo, 1969). Hence, 4 blocks of 66 pairs (=264 pairs) were presented through an Actronika HapCoil-One vibrotactile actuator (dimensions: 11.5 × 12 × 37.7 mm^3, acceleration: 8 g-pp, frequency bandwidth: 10 to 1000 Hz, resonant frequency: 65 Hz). This kind of actuator has already been used in the literature to render the sensation of textures with vibrations (Rocchesso et al., 2016). The actuator was powered by a Pioneer A-209R audio amplifier. The subjects were asked to grab the vibrotactile actuator between the thumb and the index of their right hand. During the tactile experiment, participants wore noise canceling headphones to prevent them from using auditory cues.

2.4 Tasks and procedure

In each experiment, participants were seated in front of a computer screen in a quiet room. For each subject, the pairs of stimuli were presented in randomized order. For each pair, the presentation order was also randomized. In each experiment and for

each pair of sounds, subjects were asked to judge which tone combination (or tactile stimulation) was the most “granular” (*granuleux* in French). As several participants had a musical background, we avoided the terms *rough* and *pleasant* that might also have been used. Answers were collected with a keyboard and the interface was designed with Max/MSP software to display either audio or tactile stimuli. The volume and the intensity of audio and tactile stimuli were set constant during the whole experiment.

2.5 Data analysis

In each experiment, the data were analyzed with the Bradley-Terry model (Hunter, 2004). This probabilistic model allows us to predict the outcome of a pairwise comparison from a win matrix. Practically, for each subject, a win matrix was obtained from the 66 pairwise comparisons which were sorted as follows: the cell (i,j) corresponds to the number of times the sound i has been judged rougher than the sound j. The win matrices were then aggregated between subjects and an iterative algorithm was used to fit the Bradley-Terry probabilistic model. Hence, for each frequency ratio, we obtained the probability that the corresponding combination tone was judged as rough compared to another combination. In the end, for the sake of comparison between experiments and with the literature, this probability was normalized into a perceived roughness score.

3 Results

The results, presented in Figure 2, exhibit that auditory and tactile roughness curves are very close in the 4 frequency conditions tested. Interestingly, the roughness curves obtained are also coherent with the theoretical roughness curve proposed by Vassilakis (2001). These results might suggest that auditory and tactile modalities share common principles in the perception of roughness and beatings. It would be of great relevance as it may for the first time lead to a common way between the two modalities of modeling roughness within the critical band framework. Secondly, it might further shed light on more fine similarities in the temporal processing of vibrations through these two modalities. Recent evidences have for instance shown that rhythm perception is shared between audio and haptics (Bernard et al., 2021). Our current findings suggest that these results could be extended to the perception of beating and roughness.

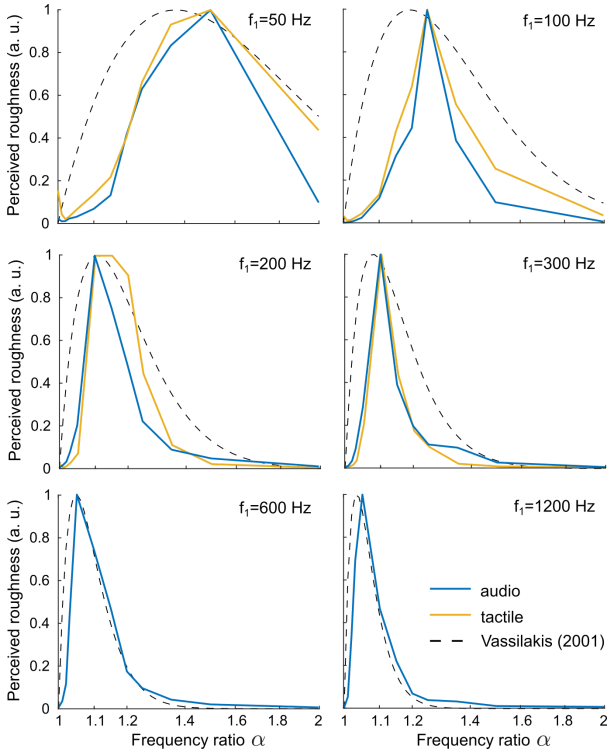


Fig. 2. Audio (blue line) and tactile (yellow line) roughness curves obtained from the experiment. The two modalities elicit similar roughness maxima and are coherent with the theoretical auditory roughness model proposed by Vassilakis (2001) (dashed line). The stimuli with f_1 = 600 and 1200 Hz were presented only for the audio condition, because they are beyond the frequency bandwidth of tactile perception.

4 Discussion

As it has already been observed in audition (Vassilakis, 2001), the position of maximal roughness perception changes according to the lower frequency f_1 also in the tactile modality. It is noteworthy that, in audition, the position of the maximum varies much more at lower frequencies, when the frequencies are inside the tactile perception range. In addition to these unimodal studies, it would be interesting to run a larger study in which subjects are asked to judge the roughness of audio-tactile stimuli. The ultimate goal is indeed to decipher and to model the way our perceptual systems combine audio and tactile senses into a coherent percept. We in particular hope to observe interactions between the two modalities and to observe how one modality may enhance the perception of roughness in the other. This has already been observed in several multisensory situations (Jousmäki & Hari, 1998, Guest et al.,

2002), and such cases are of great relevance to understand the fine mechanistic bases of human perceptual systems.

5 Acknowledgments

This work was supported by an ILCB/BLRI grant ANR-16-CONV-0002 (ILCB), ANR-11-LABX-0036 (BLRI), the Excellence Initiative of Aix-Marseille University (A*MIDEX) and the Sound and Music from Interdisciplinary and Intersectorial Perspectives (SAMI - A*MIDEX) project.

References

Arnal, L. H., Flinker, A., Kleinschmidt, A., Giraud, A. L., & Poeppel, D. (2015). Human screams occupy a privileged niche in the communication soundscape. *Current Biology*, 25(15), 2051-2056.

Bernard, C., Monnoyer, J., Wiertlewski, M. & Ystad, S. (2021) Perception of rhythm is shared between audio and haptics. *In preparation.*

Guest, S., Catmur, C., Lloyd, D., & Spence, C. (2002). Audiotactile interactions in roughness perception. *Experimental Brain Research*, *146*(2), 161-171.

Helmholtz, H. L. F. (1885). On the Sensations of Tone as a Physiological Basis for the Theory of Music (2nd edition.) Trans. A. J. Ellis. New York: Dover Publications, Inc. (1954.)

Hunter, D. R. (2004). MM algorithms for generalized Bradley-Terry models. *The annals of statistics*, 32(1), 384-406.

Jousmäki, V., & Hari, R. (1998). Parchment-skin illusion: sound-biased touch. *Current biology*, *8*(6), R190-R191.

Makous, J. C., Friedman, R. M., & Vierck, C. J. (1995). A critical band filter in touch. Journal of Neuroscience, 15(4), 2808-2818.

Plomp, R. and Levelt, W. J. M. (1965). Tonal consonance and critical bandwidth. *The Journal of the Acoustical Society of America*, Vol. 38: 548-560.

Rocchesso, D., Delle Monache, S., & Papetti, S. (2016). Multisensory texture exploration at the tip of the pen. *International Journal of Human-Computer Studies*, *85*, 47-56.

Terhardt, E. (1974). On the perception of periodic sound fluctuations (roughness). Acta Acustica united with Acustica, 30(4), 201-213.

Vassilakis, P. (2001). Auditory roughness estimation of complex spectra—Roughness degrees and dissonance ratings of harmonic intervals revisited. *The Journal of the Acoustical Society of America*, 110(5), 2755-2755.

Verrillo, R. T., Fraioli, A. J., & Smith, R. L. (1969). Sensation magnitude of vibrotactile stimuli. *Perception & Psychophysics*, 6(6), 366-372.

Towards an Aesthetic of Hybrid Performance Practice: Incorporating Motion Tracking, Gestural and Telematic Techniques in Audiovisual Performance

Haruka Hirayama[1] and Ioannis Zannos[2]

[1] Hokkaido Information University, Dept. of Information Media
[2] Ionian University, Dept. of Audiovisual Arts
hiraryama@do-johodai.ac.jp
zannos@gmail.com

Abstract. This paper discusses composing works for interactive live performance based on the comparison between recent artworks and experimental work methods of the two authors. We compare the different interaction design strategies employed and discuss the factors which influenced the choice of methods for motion tracking and their influence on body movements when coupled with the generation of sound during the performance. We consider the resulting artworks as hybrid artforms that combine aspects of music composition, improvised sound performance, and stage performance or dance. A high-level comparison of the technical and practical aspects of said works is provided. It can be argued that the new expressive potential and the wealth of possibilities to be explored warrants further work in this direction, and that systematic comparison of the interactive characteristics and expressive affordances of the systems developed are useful in guiding further research in the development of novel hybrid performance forms.

Keywords: Interactive Music Performance, Audio-Visual Art, Gesture Mapping, Telematic Art, Embodied Performance

1 Introduction

In acoustic music performance we can say that the actions or gestures of performers also provide visual cues conveying the character and shape of sound. At the same time, a performer's actions or gestures are directly coupled to the characteristics of produced sound and its musical expressive characteristics. Overall, we can say that instrumental music performance is a form of multimodal interaction synthesis [1].

The shapes of performer's gestures are dictated to a large extent by the physical properties of the instrument they are using, and appropriate techniques of performance are required to play it functionally and effectively. In addition, performers' intentions with regard to musical expression influence the form of characteristic gesture shapes. Therefore, acoustic music performance creates a fairly strict framework within which performers must stay in order to interact with their instruments in a musically effective way. The actual shape of gestures is usually regarded as playing an auxiliary role in the experience of the performance.

However, in the context of interactive computer music composition we can create new relationships between performance gestures and sound and we can choose more freely both the types of movements and the degree and type of their influence on the resulting sound. This leaves greater margins of freedom to explore the expressive potential of performance gestures from the viewpoint of their visual expressive impact. This leads to a hybrid form of expressive art that lies between visual art and music or other types of stage performance.

This paper discusses composing works for interactive live performance based on the comparison between recent artworks and experimental work methods of the two authors. The works are: *People in the Dunes,* [2] and [3] created by Haruka Hirayama in collaboration with a visual artist and choreographer Bettina Hoffmann, and *IDE-Fantasy*, created by Iannis Zannos in collaboration with dancers Jun Takahashi and Asayo Hisai (Japan) and Tasos Pappas-Petrides, Vasiliki Florou, Natali Mandila and Mary Randou (Greece) [4].

We compare the different interaction design strategies employed in the above works and discuss the factors which influenced the choice of methods for motion tracking and their influence on body movements during the performance, when directly coupled with the generation of sound during the performance. We consider the resulting artworks as hybrid art forms that combine aspects of music composition, improvised sound performance, and stage performance or dance. We discuss the degree to which system design allowing dancers to develop their individual or intuitive style of performance, with reference to the affordances created by the technical characteristics of the systems employed. Several unresolved problems arise with regard to both performance practice and the aesthetic appreciation of such works.

The extent of possible couplings of body movement to sound forms is vast, and the task of choosing or designing interaction strategies is daunting. This problem is furthermore compounded by technical limits in the accuracy and response time of movement tracking devices and by the complex, at times almost entirely unpredictable behaviour of the couplings between movement and the resulting sounds, both in terms of the physical or mathematical behaviour and from the perceptual viewpoint. However, we argue that the new expressive potential and the amount of possibilities to be explored warrants further work in this direction, and that systematic comparison of the interactive characteristics and expressive affordances of the systems developed are useful in guiding further research in the development of novel hybrid performance forms. The present paper presents a simple methodology based on a classification of the interaction techniques used in the works mentioned, and evaluating their potential based on practical factors experienced during our work.

2 The Performances

2.1 *People in the Dunes*

The *People in the Dunes* project explores expressive potential of performance with real-time sound processing as a live audio-visual art that exists at the intersection of interactive music performance and visual art involving human bodies. In this work, human

body movement plays a theatrical role while at the same time working as a medium for sound conveyance and a form of music embodiment in a manner similar to instrumental performance in music.

The *People in the Dunes* project consists of three works: *People in the Dunes I*, *The Embodiment I - Strings*, and *People in the Dunes II*. These have been created and performed in Tokyo, Montreal, and Gatineau in Canada between 2018 and 2020. The title of the project is inspired by the novel *The Woman in the Dunes* by Kobo Abe, that depicts the situation of a man trapped in the dunes fighting the ever flowing sand, reflecting about his life and in the end becoming aware of its essence and finding freedom: how human bodies and movements eventually find new directions under the influence of the forces acting from multiple directions between multiple individual actors, particularly under restricted circumstances? This project has been further developed by involving local dancers and instrumentalits working in Butoh and other contemporary styles.

2.2 *IDE-Fantasy*

The objective of *IDE-Fantasy* is to create an interactive performance which can be realized in remote locations at the same time, through the collaboration of dancers in each location, and relying entirely on motion capture data from the dancers. The piece eschews any transmission of images or sounds between the locations of the performances. The presence of the performers is transmitted between the remote stages of the performances based solely on the influence of their tracked movements on the sounds which are produced locally at each stage. Both the performers and the audience must rely on the sounds locally created by sound synthesis software to reconstruct in their imagination the actions or states of the performers in remote locations. The objective is to explore the narrative and interpretive potential of strictly reduced means for representation and the capability for sensing the states and of the performers based on the data traces left by their body movements, but without having direct visual or auditory contact.

The subject matter of the performance is inspired by the story of *Izutsu*, a Japanese Noh Play, which talks of the encounter of a monk with the ghost of a woman that is longing for reunion with her lover and husband from her previous life. Additionally, as a cultural reflection of the idea of correspondences between remote locations, symbolic correspondences between *Izutsu* and the myths of *Echo and Narcissus* and of *Daphnis and Chloe* are being explored for future realisations of this work.

The piece was developed through a series of rehearsals in Tokyo, Athens and Corfu. So far, telematic rehearsals have been realised between Athens and Corfu and Athens and Jerusalem. A performance between Stanford (USA), Athens and Corfu was presented in March 2019 at the LAC19 conference. This performance was combined with a presentation of the software framework used to create the piece [5]. A local only performance was presented at TAMA Music Festival in 2020. Further realisations are being prepared in conjunction with ongoing rehearsals and the development of new techniques for data capture and transmission.

3 Methodolody, Design Considerations, Discussion

From the perspective of an interactive music composer, the following research questions are addressed in both projects discussed: What are the expressive possibilities of music composition motivated by performance gestures? What is the theoretical framework required to create links between the shape formed by human bodies and sound, and between changes of shapes and sound transformations? Also, what are the technical means for linking physical body movement to sound production, and how can these influence the artistic process and its final outcome?

In the case of *People in the Dunes*, the experience of the composer's previous work *FRISKOTO* raised questions about the difference between performance gestures and control gestures in music. The hypothesis was posed, that the difference consists in the possibility of perceiving gestures as the animating force of music or not [1]. To make bodily movement perceivable as an animating force of music it is important to develop a system where sound can give an instant reaction to the movement, and vice versa. Furthermore, it is important to consider the correspondences between visual and auditory percepts. Visual and auditory sensations need to be properly coordinated or corresponded, in other words, their correspondences should be readily recognizable to performers as well as viewers. The following aspects guided the creative process and the design of the performances as a whole:

1. The availability of movement capture technologies and their technical performance characteristics (accuracy, reliability, temporal and measuring resolution, latency);
2. Affordances of the movement tracking devices for the performers (which movements are easy and comfortable for the performers to execute while using the tracking devices, and how do they understand the relationship between their movements and the resulting data when using the device);
3. Design of the mechanisms for influencing the sound produced based on the data received from the tracking devices;
4. Correspondence of the perceptual characteristics of the available or chosen sonic vocabulary to those of the gestural vocabulary developed by the performers;
5. Narrative effects of the sequential ordering of sequence of motions types and associated sound textures. The alternation of different motion types and types of sound textures produced by these can provide cues to the audience for understanding the causal relationship between movements and sounds, and thus aid their understanding of or identification with the performance. In addition, the ordering of motion types and sound textures can form a type of sequence of scenes that create the impression of a narrative, albeit of a fairly abstract and vague type. This plays an important role in capturing the attention of the audience, by offering hints for fabricating an interpretation of the events of the performance in their own imagination;
6. Subtle changes and minimalism. At certain parts of the performance, it is helpful to heighten the sensibility and awareness of the audience by purposely focusing on minute movements or changes of sound. This can intensify the sense of tension and the interpretive potential of the piece.

In the development of *IDE-Fantasy* we started with simple mappings between movement and synthesis parameters as few as 1 or 2, and gradually introduced more param-

eters. Even extremely simple parameter - sound mappings proved to be useful performance tools for dancers, providing them with instruments which they could explore very easily, but where nevertheless rich and responsive enough for short improvisations. In this approach, 3 or 4 parameters were already sensed as being hardly possible to handle or adapt to. 6 parameters per person were definitely outside the realm of feasible. We also experimented with 6-parameter chaotic algorithms cojointly performed by two dancers. These did capture the attention of the performers, even though they proved to be difficult to master. In table (1) we summarise the techniques used in our works.

	People in the Dunes I	*People in the Dunes II*	*IDE-Fantasy*
1. Movement tracking device	Kinect	Built-in sensors of iPhone	a) 9-axis movement b) 3-axis accelerometers
2. Number of attached sensors per person	-	2 iPhones per person	Up to 2 per person
3. Sensor positioning	-	Left and right lower arms	Wrists
4. Tracking information	Horizontal boundary position	Acceleration, Magnetic field, Gyroscope	Acceleration, Magnetic field, Gyroscope
5. Data transmission protocol	USB	WiFi	WiFi, Xbee Mesh Network
6. Software(s) for interactive systems	Max, Jitter	Max, ZIGSIM (iOS)	SuperCollider
7. Audio source for composition	a) Boundary microphones b) Prerecorded voice sound	Prerecorded voice, cello, traffic, environmental sound, synthesisers	a) Prerecorded samples b) Simple or Complex UnitGenerator graphs with or without feedback

Table 1. Technologies used in *People in the Dunes* and *IDE Fantasy*

4 Conclusion, Future Work

In all, a common trait observable in both works discussed here is the use of the technical affordances of motion tracking devices and sound generation or processing algorithms to design a sort of performance language which combines body movements and their assigned sound textures or events to create narratives of a more or less abstract type. In both pieces, concrete narratives of previously existing and well known works provided a reference framework in order to create the more abstract narrative of the pieces.

In conclusion, the main challenges confronting this kind of work stem from the abstract and indirect nature of digital mediation between bodily movement and generated audiovisual stimuli. The causal relationship between movement and generated sound or video tends to be difficult to recognise. In some cases it can be entirely absent, as is when employing chaotic synthesis algorithms. To counterbalance these obstacles, it

is necessary to create interpretive or narrative links with the performers and the audience. Simple mappings and complex or chaotic correspondences both present advantages and disadvantages, and the decisive design criteria for developing a functioning performance seem to lie in semiotic domains such as the choice of sounds, images, and movements for their associative semantic charge, and the devising of narrative devices through trial and error during rehearsals. Currently we are interested in employing Machine Learning algorithms in order to devise improved methods for translating motion to sound, and in particular in experimenting with unsupervised learning and in adaptive techniques that modify their behaviour during the performance itself.

At the same time, the characteristics and affordances of motion tracking devices (shapes of sensors, kind of detecting data, mobility etc.) have a direct impact on the available movements and thereby on the kind of body movement language that the performers develop. We feel that a combination of different type of tracking method can enhance performance expression.

The body is capable of constantly adapting and changing shapes or forms [6]. We realised that there is an interdependence between isolated movements of individual parts of the body and the perception of forms created by movements of the body as a whole. This will serve as a guiding principle for the currently planned experiments for future work.

References

1. Collins, Karen, Bill Kapralos, and Holly Tessler, eds. The Oxford Handbook of Interactive Audio. Oxford University Press, Oxford (2014).
2. People in the Dunes project page, `https://www.facebook.com/peopleinthedunes/`, last accessed 2021/06/15
3. People in the Dunes II videos, `https://vimeo.com/user88406194`, last accessed 2021/06/15
4. IDE-Fantasy videos, `https://ide-fantasy.tumblr.com/`, last accessed 2021/06/15
5. Zannos, I.: SC-Hacks: A Live Coding Framework for Gestural Performance and Electronic Music. In: Proceedings of the 2019 Linux Audio Conference. pp. 121-128. CCRMA, Stanford University (2019).
6. Kobayashi, Y., Matsuura, H., eds. The Discourse of Représentation Skinship: The Rhetoric of the Body. University of Tokyo press, Tokyo (2000).

Evaluating AI as an assisting tool to create Electronic Dance Music

Niklas Bohm[1], Christian Fischer[1] and Manuel Richardt[1]

Hochschule Fulda
christian.fischer@ai.hs-fulda.de

Abstract. The demands on creatives to complete a jingle or a piece of music even under time pressure are growing. This paper analyzes Google's "Magenta" to identify its possibilities for a more effective production of electronic dance music *(EDM)*, especially in terms of time, without a loss of subjective listening pleasure. For this purpose, the process of EDM music production, which includes artificial intelligence, was analyzed. With a subsequent survey, it was determined whether the music pieces produced in this way differ in their subjective listening pleasure and which of the approaches can be recommended for further production.

Keywords: Google Magenta, Electronic Dance Music, AI, Music Production

1 Introduction

Artificial intelligence is used more and more in everyday life and therefore discussed controversially in media. In this text AI is understood as a digital tool to support humans in their creative tasks. However, when one reads about AI in general, some authors find it obviously still difficult to think of AI in connection with creative tasks [6]. There are reservations of creative processes being mapped by a computer [8]. Music is one such creative construct, cause it is capable of triggering feelings in people and amplifying or even influencing moods. Interestingly, most listeners don't notice how much music composition is already influenced by AI and that nowadays many artists have already introduced AI into their creative production work flow [3]. Still, it is not a question about whether AI will make the human composer obsolete, but how creative people use AI in their process [9]. Therefore, the aim of this work is to find out how the use of AI can affect the efficiency of the music production process and whether it has an influence on how the music is perceived. The main focus was efficiency in time and to figure in which steps within the production, in the case of Electronic Dance Music *(EDM)*, an AI can be particularly helpful.

2 Related Work

There is an area of application for artificial intelligence in almost every field. Among others, also in the creative areas of music and art. Because AI has so many forms, researchers divide it into the subfields of automation, machine learning, neural networks, and deep learning [2]. A common form of neural network is the Convolutional Neural

Network *(CNNs)*. In this paper, and the Magenta project respectively, Recurrent Neural Networks *(RNNs)* and the Long Short-Term Memory Network *(LSTM)*, a type of RNN, are central [11].
While CNNs tend to be hierarchical, RNNs operate more sequentially [1]. Therefore, CNNs are often used for classification tasks such as text recognition, whereas RNNs are more flexible and are therefore increasingly used for language processing and creative tasks such as music generation [11]. The project "Magenta", placed under Open Source by Google is a project which explores machine learning in a creative context. Magenta Studio, basically a collection of music plugins, uses RNN and LSTM networks. To understand how a computer can utilize this to make music, it is important to take a closer look at these RNNs. An RNN is a class of neural networks used to model sequence data. In an RNN, the connections between artificial neurons form a circuit. For many applications such as speech and text, outputs can depend on previous inputs and outputs. The key concept of RNNs is to use sequential information. RNNs are thus given a "memory" in the form of the information and data that have been processed so far [1]. Related to the music context in this paper, this means that the neural network can, e.g. recognize which notes it has generated in previous measures and is able to adjust the next measure to sound coherent. The size of the "memory" determines how many notes, or even bars the neural network can look back. The goal is to have the generated sounds or musical sequences which contain repetition, as repetition is one core feature in music.

3 Environment Setup and AI-Created Music

Magenta is the name of several research projects by Google. The program Magenta Studio, serves as AI support for digital music producers, as a standalone application or when used as a Plugin e.g. with the Digital Audio Workstation Ableton Live. Magenta has five different models, using different neural network types. In this study LSTM networks, a type of RNN, are used. An example of an LSTM network is the DrumsRNN model. Another example is the MelodyRNN model, which has the same structure but is programmed for melody generation [4]. Each model has different configurations that set the way in which it encodes the input data. For example, the DrumsRNN model has a one drum configuration, which stores a drum sequence in a class, and a drum kit configuration, which splits a drum sequence into nine different instruments (kick, clap, hi-hat, etc.) and adjusts the attention length [4]. Magenta provides many pretrained models for download on its site, here the models "DrumsRNN", "MelodyRNN" and "PolyphonyRNN" were used.
To structure the process of digital music production and to make the influence of AI in this process more comprehensible, we divided it into three levels. The higher the level, the less human involvement in the production process.
Level 1 - Inspiration: In this stage, the human producer is supported or inspired by the AI. For example: AI generated melodies are listened to. Being inspired, the human producer composes a new melody which contains no or hardly any parts of the original AI melody.
Level 2 - AI Assistance/Co-production: In AI assistance, one gets support from an AI

in the creative process. Here, bigger parts of AI-generated music can be incorporated into the composition. The AI can be used to help, e.g. by suggesting subsequent notes. Since the AI and human are both working on the same piece, they can both inspire each other.
Level 3 - Automation: In automation, an AI generates new music on its own without input. The producer has no influence on the results. One of many examples of autonomous music generation would be Magenta's Generate function [10]. One other example is AIVA, which stands for Artificial Intelligence Virtual Artist. AIVA has already released self-produced albums in the EDM genre. It was created in early 2016 and has been under constant development ever since [5].

4 Tests Structure and Efficiency Evaluation

First three simple EDM tracks, according to the three stages, each consisting of the three main components bass, chords and melody, were composed. For the later evaluation, some measurement data had to be collected. Thus, the production duration and effort were measured based on mouse movement, mouse clicks and key usage, utilizing the free software Mousotron (Win; version 12.1; Black Sun Software 2017). In order not to confuse the participants of the later survey by different drums and drum patterns in the songs, an identical one was adopted here for all. For the music production itself the Digital Audio Workstation Studio FL (Win; version 20.8; Image-Line Software; 2020) was used. One aim was to verify whether the introduction of AI into the production process is suitable for speeding up and simplifying music production. For this purpose, the individual steps of the production were calculated, and all mouse clicks and movement were measured as the second and third measured values.
The evaluation (collecting data of mouse and keyboard usage) was carried out through the different phases to determine and track the highest increase in efficiency on specific production phases and production steps. In this way, phases for which AI is particularly suitable can be identified (see also Table 1).
Preparation phase: In the preparation phase, human production performed best. Preparation took just under a minute, with both AI automation and AI assistance at around three minutes.
Composition phase: In the composition phase, the tendency of the first phase changes. Here, human production took by far the longest at around nine minutes. With the help of AI assistance and AI automation, up to 85% of the time used could be saved.
Editing phase 1 (without mixing and sound design): In this phase, it is noticeable that the AI assistance has particularly high values. To adjust an AI given melody to one's own desires might take time. The duration of the production phase was therefore about four to five times longer for the AI assistance than for the other two production types.
Editing phase 2 (mixing and sound design): Great inspiration can come from sound design, as melodies can also sound very different depending on the sound. Since a producer usually creates a melody after the sound design, this phase also takes about twice as long as the other production types. With the support of AI, one comes to a time saving of about 65% here. The AI assistance has the lowest effort in this phase, but this could change if the experiment is performed multiple times.

Human production took the longest in terms of total production time. However, this is closely followed by AI assistance. The use of AI automation reduced the duration of production by 40%. For composition only (without processing phase 2), there was even a 42% reduction. AI assistance took about 23% longer than a human production in the pure composition process.

Phase	Song	Duration	Mouse clicks	Mouse distance	Keystrokes
Preparation	Human production	01:12	22	3,51	24
	Automation	03:20	64	18,56	76
	AI-Assistance	02:52	53	15,45	57
Composition	Human production	09:07	203	20,57	239
	Automation	01:43	16	1,41	12
	AI-Assistance	01:23	16	1,56	14
Editing 1	Human production	02:34	41	5,87	12
	Automation	03:02	36	7,19	95
	AI-Assistance	13:01	227	30,28	433
Editing 2	Human production	07:48	141	21,34	78
	Automation	03:47	67	12,17	67
	AI-Assistance	02:34	40	8,41	52

Table 1. Measurements in different production phases

5 Listening Evaluation

The second aim of this study was to find out how listeners like the generated songs and whether they can tell the difference between an AI-generated and a human production. To investigate these questions, 50 participants with different musical backgrounds and individual tastes (in balanced proportions) were asked to do an online questionnaire stating their opinions about the three generated songs.

After general questions (age, gender, knowledge/involvement in music and knowledge of EDM), the participants listened to the songs and were asked to rate the songs based on the following eight evaluation criteria: Creativity, Recognition Value, Arrangement (of a song), Cohesiveness, Variety of Tones, Energy, Danceability, and Emotion. Through a study conducted by the University of California Irvine, these criteria were identified as crucial. The study found that songs are especially popular when they are more upbeat and danceable than other songs [7]. In addition, dynamics and the mood of the listeners play a role [7]. Other criteria such as the vocals or the genre are not considered in this paper it focuses exclusively on instrumental EDM music.

The participants could rate the different criteria from 1 to 5 points. One meaning low and five meaning high. All points from all 50 participants that were given for one song were added to an overall score for that song (see Table 2).

Song 1 - Automation: The first song was generated by the AI alone. The creativity of the song was rated an average of 3 and is therefore exactly in the middle of the scale. The recognition score averaged 2.7 points and the arrangement was rated at 2.76 points

Song	Creativity	Recognition Value	Arrangement of tones	Cohesiveness	Variety of Tones	Energy	Danceability	Emotion
1	3	2.7	2.76	3.22	3.19	3.8	3.4	2.47
2	3.39	3.43	3.8	3.76	3.27	3.43	3.65	3.14
3	3.82	3.58	3.56	3.39	3.37	3.66	3.37	2.98

Table 2. Results of the questionnaire

on the scale. For the AI-generated song, energy got the highest score with an average of 3.8 points on the scale. This was followed by danceability with 3.4 points and cohesiveness with 3.22. Emotion was rated the worst with a mean of 2.62 points. Overall, Song 1 collected 1228 points in all criteria by all participants or an average value of 3.08 points on the scale.

Song 2 - Human production: The second song was produced by humans. The arrangement scored best with an average of 3.8 points on the scale. This was followed by cohesiveness with 3.76 points and danceability with 3.65 points. Emotion was again rated the worst with a mean of 3.14 points. Overall, the song was rated with 1365 points and achieved an average value of 3.49 points on the scale.

Song 3 - AI assistance: The third song was produced in collaboration between a human and an AI. Creativity received the highest score here, with an average of 3.82 points on the scale. This is followed by energy with 3.66 points and recognition with 3.5 points. Emotion was again rated the lowest with a mean score of 2.98 points. Overall, the song was rated with 1368 points and achieved an average value of 3.42 points on the scale.

In comparison, the human-produced song, and the AI-human collaboration both scored about the same. The AI-generated song scored 1228 points, about 10% worse than the other songs. Creativity is rated the highest for the AI-assisted production, and the recognition value is also almost 30% higher. The human production was considered less creative, but the arrangement was easy for the listener to understand, and the song seemed cohesive. In the AI production, the rhythmic criteria such as danceability and energy stood out.

6 Summary and Conclusion

The experiment was designed to test whether the use of AI in digital music production can increase the time efficiency. In the experiment, three songs were produced with the same prerequisites, but each in a different production mode. Since AI can be used to different degrees in music production, three levels were developed: AI inspiration, AI assistance, and AI automation. As a result, it was expected that the producer will have significant savings in time and effort by using AI. However, some of the results differed from the assumptions. A digital music production was divided into four phases. The experiment showed that the use of AI in production increased efficiency especially in the composition phase and the second editing phase. There were no significant differences in the overall production ratio in the preparation phase, but this phase was slightly faster with human production. Overall, the AI automation was convincing with an efficiency increase of 41% on average. AI assistance decreased production efficiency by about

20% on average.
The created songs from the experiment were listened to, rated, and evaluated by 50 people. The result of the survey was that the computer-generated song was rated about 10% less, than the other two songs. It is concluded that although an AI can write songs and melodies on its own, they are perceived not as melodic and creative as productions that involved a human. Since AI assistance was rated 30% better in terms of melodic criteria, it can be stated that the introduction of AI into the production process can exceed these very limits.
Thus, for the further development of AI-assisted programs for music production, it is important to develop AIs that see a song as a whole and when the "memory" (e.g., in LSTM) is large enough to remember a theme or idea in a song and repeat it at the right places. For EDM producers, it should be noted that the introduction of AI as a tool into the right phase of the production process has many positive aspects such as saving time and effort and supporting the creative process of composing melodies.

References

1. Bisharad, D., Laskar, R. H.: Music genre recognition using convolutional recurrent neural network architecture. Expert Systems 36 (4) (2019)
2. Dargan, S., Kumar, M., Ayyagari, M.R., Kumar, G.: A Survey of Deep Learning and Its Applications: A New Paradigm to Machine Learning. Archives of Computational Methods in Engineering 27 (4), 1071–1092 (2020)
3. Deahl, D.: How AI-generated music is changing how hits are made. The Verge, https://www.theverge.com/2018/8/31/17777008/artificial-intelligence-taryn-southern-amper-music (2018)
4. DuBreuil, A.: Hands-on music generation with Magenta (E-Book). Explore the role of deep learning in music generation and assisted music composition. Birmingham, Packt (2020)
5. Hewahi, N., AlSaigal, S., AlJanahi, S.: Generation of music pieces using machine learning: long short-term memory neural networks approach. Arab Journal of Basic and Applied Sciences 26 (1), 397–413 (2019)
6. Holeman, R.: Why A.I. Can't Replace Human Creativity. Modus Medium, https://modus.medium.com/when-it-comes-to-creativity-is-it-game-over-for-humans-ef12907eb30d (2019)
7. Interiano, M., Kazemi, K., Wang, L., Yang, J., Yu, Z., Komarova, N.L.: Musical trends and predictability of success in contemporary songs in and out of the top charts. Royal Society open science 5 (5) (2018)
8. Müller, V.E.: Risks of Artificial Intelligence. In: Vincent C. Müller (Hg.). Risks of artificial intelligence. Boca Raton, FL, CRC Press, 1–8 (2020)
9. Pennington, A: Artificial Intelligence gets creative. IBC, https://www.ibc.org/how-ai-is-being-used-to-boost-the-creative-process/2937.article (2018)
10. Roberts, A., Engel, J., Mann, Y., Gillick, J., Kayacik, C., Nørly, S., Dinculescu, M., Radebaugh, C., Hawthorne, C., Eck, D.: Magenta Studio: Augmenting Creativity with Deep Learning in Ableton Live (2019)
11. Yin, W., Kann, K., Yu, M., Schütze, H.: Comparative Study of CNN and RNN for Natural Language Processing (2017)
12. Mousotron, https://mousotron.de.softonic.com/
13. Studio FL, https://www.flstudioshop.de/fl-studio-/

WaVAEtable Synthesis

Jeremy Hyrkas*

University of California San Diego
jhyrkas@ucsd.edu

Abstract. Timbral autoencoders, a class of generative model that learn the timbre distribution of audio data, are a current research focus in music technology; however, despite recent improvements, they have rarely been used in music composition or musical systems due to issues of static musical output, general lack of real-time synthesis and the unwieldiness of synthesis parameters. This project proposes a solution to these issues by combining timbral autoencoder models with a classic computer music synthesis technique in wavetable synthesis. A proof-of-concept implementation in Python, with controllers in Max and SuperCollider, demonstrates the timbral autoencoder's capability as a wavetable generator. This concept is generally architecture agnostic, showing that most existing timbral autoencoders could be adapted for use in real-time music creation today, regardless of their capabilities for real-time synthesis and time-varying timbre.

Keywords: Generative models, neural networks, sound synthesis

1 Introduction

A generative model can be broadly defined as a probabilistic method that learns a distribution based on a corpus of training data such that examples similar to the training data can be generated by sampling from the learned distribution [1]. Recently, the term has been largely associated with deep artificial neural networks that generate images, video, speech, or examples from a variety of other domains. Music researchers have utilized neural network generative models as a technology for sound synthesis in music (for example, the groundbreaking NSynth neural synthesizer [2]). One such approach is the *timbral autoencoder* (i.e. [3][4][5]). In this type of model, networks learn audio representations in the frequency domain, resulting in models that synthesize sounds based on a learned latent space of their training data, usually monophonic instruments. These timbral models target the problem of novel sound generation, particularly in a synthesizer setting [3]. Ideally, musicians can find sounds that interpolate the timbre of multiple instruments, or sounds that do not invoke any recognizable instrument at all. Recently, the variational autoencoder (VAE) [6] has been favored (an overview of VAEs for musical audio can be found here [1]). Once trained, a user may provide latent parameters to the VAE to generate new examples.

There are a number of benefits to training these models. The training data is represented in the frequency domain, which behaves better than time-domain representations with common loss functions that do not account for phase shift. Additionally,

* Special thanks to Karl Yerkes (MAT, University of California Santa Barbara) for his great help with SuperCollider and OSC implementations.

some models render audio in near real-time [3] due to a sufficiently small architecture, as opposed to more complex audio models such as the NSynth autoencoder [2]. More recent efforts have shown that the models can learn the timbre of the training data in a way that sufficiently disentangles the pitch of the training examples, even when the training data does not contain pitch labels [4].

While integration of these models in music systems seems imminent, there are some practical drawbacks that remain unaddressed. Learning problems in the frequency domain largely concern magnitude spectra, meaning that phase reconstruction is necessary before the audio can be rendered in the time domain. Models that learn on magnitude Fourier transforms can use phase reconstruction algorithms that run in real-time [3]; however, models that use alternate spectral representations [5] rely on non-real-time algorithms such as the Griffin-Lim method [7]. Finally, recurrent connections are largely absent in timbral autoencoder models, meaning models are limited to generating a cyclic waveform per selection of latent parameters.

1.1 Motivation and Project Overview

This project aims to integrate a neural network synthesis engine implemented in Python with more general synthesis and composition engines in Cycling '74's Max software and SuperCollider (SC). While many of the aforementioned research efforts focus on improved synthesis in the form of fidelity, realism, or expressiveness, this project takes the philosophy that current synthesis methods are already usable in music creation when combined with existing and well understood computer music methods.

Because most timbral autoencoders produce inherently cyclic audio, we can conceptually treat them as oscillators. Many timbral autoencoders are not conditioned on pitch, precluding them from being used as oscillators in a traditional sense. Additionally, models that do not use real-time phase reconstruction cannot be used to synthesize audio in real-time like a traditional oscillator. Therefore, incorporating these models into real-time engines requires a method that utilizes pre-rendered cyclic audio signals.

The most straightforward candidate for such a synthesis system is wavetable synthesis [8], a scheme in which cyclic waveforms are stored and synthesized by reading and interpolating values at a given frequency. This project recasts timbral autoencoders as wavetable generators and provides methods for sampling and saving wavetables from their output. Proof-of-concept software is provided in Max and SC, demonstrating how timbral autoencoders as wavetable generators can be used in performance and composition, and can be sonically extended using methods such as wavetable interpolation and frequency-modulated playback. We refer to the process of incorporating timbral autoencoders, often VAEs, into a wavetable extraction framework and combining them with music synthesis software as WaVAEtable synthesis.

1.2 Related Methods

NSynth [2] is an early musically focused generative model for audio. NSynth's unique architecture allows it to iteratively create time-domain audio, resulting in audio with time-varying timbre. This result is arguably more musically useful than cyclic waveforms, but the model is expensive to train on most computers and slow to render audio.

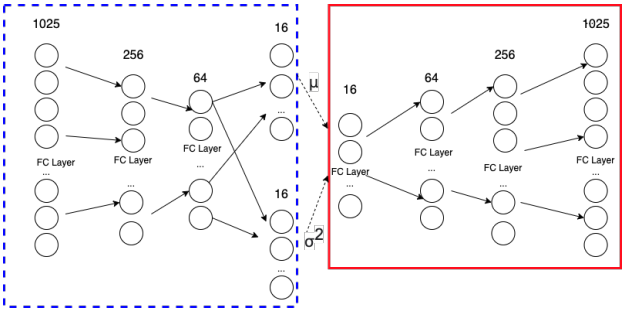


Fig. 1. Architecture of the basic VAE trained for this project. The *dotted blue rectangle* on the left contains the encoder and *solid-lined red rectangle* on the right contains the decoder. The *dotted line* between encoder and decoder represents a sampling operation. The input and output data are magnitude spectra extracted from the STFT of instrumental audio.

Neural Wavetable [9] is perhaps the most similar project to the one presented here. The project uses an autoencoder to learn time-domain wavetables and interpolate between them. Interpolation is performed on the latent encoding of two target wavetables. This is conceptually similar to timbral autoencoders, with the exceptions that Neural Wavetable operates in the time-domain as opposed to the frequency-domain, and that the model is explicitly trained on wavetables. Because the method proposed here extracts wavetables from a broad collection of generative models, it is a more general method than the Neural Wavetable method. Neural Wavetable's underlying model cannot generate audio in real-time, so the associated plug-in uses pre-rendered wavetables for interpolation.

A more thorough survey of generative models for audio can be found here [1].

2 WaVAEtable Synthesis

2.1 Sample neural network architecture and training

This software exploits the architecture of timbral autoencoders, wherein user-provided encodings produce spectral audio that is converted to time-domain audio. To keep the design aimed towards utilizing existing models, this software uses a simple VAE (depicted in Figure 1) implemented in PyTorch [10] that encapsulates the most basic generative capabilities shared by timbral autoencoder models. The data are positive frequency bins from Short-time Fourier Transform (STFT) frames of audio in the NSynth dataset [2]. The architecture model is shown in Figure 1. After training, the 16 latent parameters are used to synthesize the magnitudes of the positive frequencies of an STFT frame. These frames are reflected and time-domain audio is added using the Griffin-Lim algorithm [7]. The decoder-to-audio process is detailed in Figure 2. Adjusting the architecture and hyperparameters of this model could constitute a separate research effort, but are not critical to this project as this method aims to be as architecture- and model-agnostic as possible.

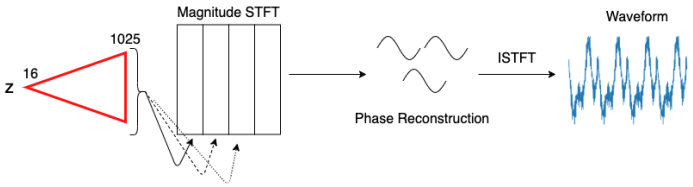


Fig. 2. Decoder to audio process: latent space parameters (z) are provided to the decoder, depicted as a *red triangle*. Output from the fixed z is used to create a *magnitude STFT*. The Griffin-Lim algorithm for *phase reconstruction* then yields in time-domain audio.

2.2 Wavetable generation

Many synthesis engines use a default wavetable size of 512 samples. If a given timbral autoencoder is conditioned on pitch, the model could generate output at a specific fundamental frequency f_0 such that the period of the waveform is 512 samples given the sampling rate f_s by setting $f_0 = \frac{f_s}{512}$. However, this setup is not always feasible. Many timbral autoencoder models, including the model used here, are not conditioned on pitch, resulting in a generative latent space that changes both timbre and fundamental frequency simultaneously. Even those models that disentangle pitch from timbre may be conditioned on discrete pitches (such as MIDI notes), and in general may not be able to generate the desired f_0. For example, the sampling rate of the NSynth data set is 16 kHz, so a waveform with period 512 samples has $f_0 = 31.25$ Hz, well below reasonable pitches in most music data sets.

Therefore, wavetables are created using a heuristic wavetable extraction algorithm that relies on f_0 estimation and resampling. Given a latent encoding from a user, the decoder is invoked to create a periodic waveform (see Figure 2). We use the pYin [11] algorithm to estimate the fundamental frequency of each frame. The pYin algorithm is probabilistic and determines the likelihood of each frame containing a pitched sound. If a sufficient number of frames are found likely to be pitched, we predict the f_0 of the waveform to be the mean of the f_0 of the voiced frames; if this is not the case, it is likely that the provided encoding is very dissimilar from examples learned in the training data and the resulting sound may be noisy and therefore unpitched.

Given the f_s of a model and the predicted f_0 of the model's output based on the user's inputs, we resample the output to a new sampling frequency $round(f_0 * 512)$, which results in a waveform whose period is very close to 512 samples. Finally, samples are extracted from the new waveform starting from some position that is very near 0 to avoid an impulse at the beginning of the wavetable. Overall, the extraction method is subject to failures in f_0 estimation (usually octave errors) and resampling artifacts, but is architecture agnostic and can be adapted to any timbral autoencoder (or any generative model whose output is sufficiently periodic).

2.3 Synthesizer implementations

Two prototype synthesizers were created as a proof-of-concept to demonstrate multiple musical uses for VAE wavetables. First, a simple polyphonic synth patch was created

in Cycling '74's Max software. This patch assigns incoming MIDI notes to one of five voices. Here, random wavetables are pre-rendered using a Python script and can be regenerated by the user. Communication is done via the file system with wavetables saved and loaded from .WAV files. This patch also combines wavetable synthesis with FM synthesis, with MIDI CC controls controlling the wavetable assignment and FM controls of the wavetable playback speed. This patch, while simple, demonstrates the viability of using timbral autoencoder output in real-time performance.

A more complex system was constructed in SC with the goals of user interactivity with the underlying model, more complex synthesis methods and algorithmic composition. Users control the latent parameters of the VAE described in Section 2.1 using a custom GUI created in SC. The user can listen to a wavetable for a given setting, and if it is interesting for their compositional purposes, save it. All communication between Python and SC is performed locally using OSC, so no file system interaction is required. Figure 3 shows the interface for manipulating and storing wavetables.

Once stored, wavetables are played back using wavetable synthesis and wavetable interpolation. Users can also incorporate other SC generators to create complex synthesizer definitions (SynthDefs) with the generated wavetables at their core. We provide an example SynthDef that can be controlled by a MIDI controller, or used in algorithmic composition. A small etude is included in the provided software to demonstrate this capability. All Max, SC, and Python code, as well as the accompanying VAE model, are available at `https://github.com/jhyrkas/wavaetable`.

2.4 Incorporation of existing timbral autoencoders

The neural network used in this project is not intended to be a standalone model, but acts as a basic stand-in for existing timbral autoencoder models (i.e. [3][4][5]), most of which have more complex architectures and are capable of more pleasing musical audio. To test the viability of the WaVAEtable synthesis approach, the Python script to interface with SC was adapted and added to a fork of the CANNe [3] synthesizer GitHub, available at `https://github.com/jhyrkas/canne_synth`. The only major changes to the script involved reinterpreting the latent space parameters sent from SC, as CANNe's latent space only contains 8 variables and expects a different range of values. With just these minor adjustments, the CANNe model can now be used as a wavetable generator in WaVAEtable synthesis. We posit that other timbral autoencoders can also be easily adapted, so long as they offer an encoding-to-audio synthesis method.

3 Future Work and Conclusion

This work offers a path towards incorporating an existing body of generative models into music systems. The proposed method allows for integrating models regardless of underlying architecture and real-time viability, and allows for a greater reuse of interesting latent parameters, which can be cumbersome to discover. Synth design and model improvement can thus be treated as complementary and orthogonal research avenues.

WaVAEtable synthesis may approach the practical limits of incorporating static generative models for audio in more traditional electronic music synthesis. Future timbral

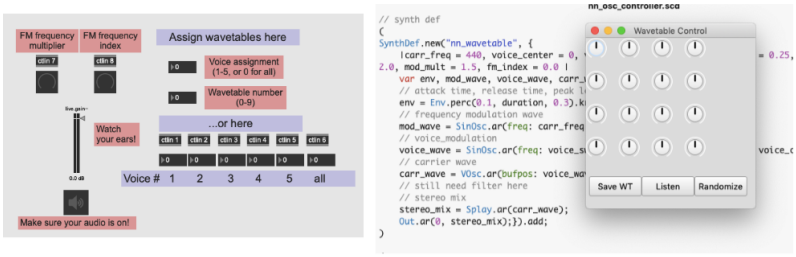


Fig. 3. Left: Max interface to playback wavetables, controlled via MIDI controller. Right: SuperCollider interface to control decoder parameters, listen to and store wavetables for playback.

models that generate audio in real-time and are conditioned on pitch could function as a true oscillator in a synthesis system. Moving beyond these static timbral models to time-varying models allows for new combinations of generative models and synthesis methods, such as neural sample-generation and neural granular synthesis.

References

1. Huzaifah, M., Wyse, L.: Deep Generative Models for Musical Audio Synthesis. In: Handbook of Artificial Intelligence for Music: Foundations, Advanced Approaches, and Developments for Creativity. Springer (2020)
2. Engel, J., Resnick, C., Roberts, A., Dieleman, S., Norouzi, M., Eck, D., Simonyan, K.: Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70 (2017)
3. Colonel, J., Curro, C., Keene, S.: Autoencoding Neural Networks as Musical Audio Synthesizers. In: Proceedings of the 21st International Conference on Digital Audio Effects (2018)
4. Luo, Y. J., Cheuk, K. W., Nakano, T., Goto, M., Herremans, D.: Unsupervised Disentanglement of Pitch and Timbre for Isolated Musical Instrument Sounds. In: Proceedings of the 2020 International Society of Music Information Retrieval Conference (2020)
5. Esling, P., Chemla–Romeu-Santos, A., Bitton, A.: Generative timbre spaces: regularizing variational auto-encoders with perceptual metrics. In: Proceedings of the 21st International Conference on Digital Audio Effects (2018)
6. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. In: arXiv preprint, arXiv:1312.6114 (2013)
7. Griffin, D., Lim, J.: Signal Estimation from Modified Short-Time Fourier Transform. IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, num. 2, 236–243 (1984)
8. Bristow-Johnson, R.: Wavetable Synthesis 101, A Fundamental Perspective. In: Proceedings 101st Convention of the Audio Engineering Society (1996)
9. Hantrakul, L., Yang, L. C.: Neural Wavetable: A Playable Wavetable Synthesizer Using Neural Networks. In: Workshop on Machine Learning for Creativity and Design (2018)
10. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: Proceedings of the 31st Conference on Neural Information Processing Systems (2017)
11. Mauch, M. Dixon, S.: pYIN: A Fundamental Frequency Estimator Using Probabilistic Threshold Distributions. In: Proceedings of the 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (2014)

With Love: Electroacoustic, Audiovisual, and Telematic Music

Paulo C. Chagas[1] and Cássia Carrascoza Bomfim[2]

[1] University of California, Riverside
paulo.chagas@ucr.edu

[2] University of São Paulo, Ribeirão Preto
cassiacarrascozabomfim@usp.br

Abstract. This article discusses different approaches of music composition and performance with electroacoustic, audiovisual and telematics media. It provides different points of view for understanding the so-called electroacoustic paradigm which emerges from the use of apparatuses in the sound creation and production. From within electroacoustic music paradigm, we examine tendencies and visions of audiovisual and telematic music composition and performance. As illustrations we examine the pieces *Vega_S* (2019) by Kefalidis and *Mojave* (2021) by Chagas/Carrascoza. The telematic communication has the potential do convert discursive thinking into dialog and opens up new possibilities of artistic collaboration. The holistic potentiality of telematic art supports Ascott's metaphor of love in the telematic embrace.

Keywords: electroacoustic music, audiovisual music, telematic music, composition, performance, artistic collaboration, Flusser, Ascott.

1 Electroacoustic Music

What is electroacoustic? And what is electroacoustic music? From an evolutionary perspective, electroacoustic music represents a new paradigm in the history of music that carries on the tradition of vocal and instrumental music and extends it to include the use of apparatuses to produce and move sound around in spaces. From this historical point of view, it has emerged in a period of crisis represented by the disruption of the fundamental role tonal harmony has played as the established disciplinary matrix of music composition. This crisis triggered different responses leading to non-tonal textures in the music of composers such as Schoenberg, Webern, Stravinsky, Debussy, Bartok, Messiaen, and others. Moreover, it pushed composers to explore other constructive principles of musical organization focused on the physical reality of sound phenomena, and to emphasize sound qualities such as timbre and noise. Within the crisis of tonality as foundation, electroacoustic music was able to meet the demands of an aesthetic sensibility focused on this expanded consciousness of sound phenomena.

We find three different orientations in the development of electroacoustic music: *musique concrète*, *elektronische Musik,* and computer music.[1]

The *musique concrète* that came into existence in Paris after World War II, began with Pierre Schaeffer's experiments in recording techniques for capturing sounds of the acoustic environment. This approach engaged the persistent myth that the world is the primary acoustic space of music extending from the earth to the whole universe. The acoustic myth allows sound phenomenon to be isolated from the physical environment, be heard as a unique object and event, and eventually be disconnected from its material source and origin. Released from its cultural references, sound becomes a self-referential paradigm for composing new audible forms. At this point, composition took advantage of new technology for recording, manipulating, and reproducing sound. Drawing ideas from Edmund Husserl's phenomenology of time consciousness [2], the aesthetics of *musique concrète* developed notions such as *sound object* and *reduced listening*. These categories emerged through the interaction of sound material with technical apparatuses, most notably, the tape recorder. *musique concrète* provided electroacoustic composition with analytical and synthetic approaches to sound perception and composition.

The *elektronische Musik*, most closely associated with the electronic music studio of Cologne, pioneered the creation of sounds whose models are neither found in nature nor possess the qualities of instrumental or vocal sounds. Methods adopted by Karlheinz Stockhausen and other composers of *elektronische Musik* were used to invent new sounds building from the simple elements of technical apparatuses. The signal generator and the noise generator became the prototypes of electronic sound devices despite being designed to test equipment and not for making music. These apparatuses are both mathematical constructs; the signal generator explores the simplicity of a single harmonic motion such as the sine wave, while the noise generator explores the statistical model of all possible vibrations occurring randomly in auditive space. The aesthetics of *elektronische Musik* took advantage of electroacoustic technologies developed during the German Third Reich, which radically transformed the experience of listening while creating new logics to frame political activity. Radio broadcasting and sound amplification were interconnected technologies used for acoustic landscape control and organic synchronization of masses. Radio in particular activated the sonic experience of private intimacy and transformed the universe of telematic paradigm. However, radio also preserves the ancient magic of mythical worlds. As McLuhan [3, 299] notes, "The subliminal depths of radio are charged with the resonating echoes of tribal horns and antique drums."

The historical opposition between *musique concrète* and *elektronische Musik* is emblematic of the diversity of the electroacoustic paradigm. After World War II, the activity of cultural institutions such as the radio studios of Paris and Cologne, promoted a shift of consciousness in electroacoustic music composition. *musique concrète* developed a poetics of *detachment* from the previous vocal and instrumental paradigms and *attachment* to the sound phenomenon; it disengaged sound consciousness from the models of traditional vocal and instrumental music while at the same time, moved

[1] For an account of the development of electroacoustic music see [1, 103-158].

toward interactions with sound that revealed cultural values and identities. Meanwhile, *elektronische Musik* developed a poetics of *detachment* from the sound and *attachment* to the paradigm of music composition. By carrying on the compositional path of the previous vocal and instrumental paradigms, it disentangled consciousness from the representative background of sound as a meaningful artifact and focused on the musical relevance of sound phenomenon. *Elektronische Musik* explores differentiations of acoustical agency in the vibration-centered model of sensitivity.

The *heterogeneity* of sound material is an aesthetic foundation of electroacoustic composition. The opposition between recorded sounds (*musique concrete*) and synthetic sounds (*elektronische Musik*), quickly dissipated as any kind of sound could become the object of musical composition. The electroacoustic paradigm not only integrated the musical puzzles of the previous vocal and instrumental paradigms but provided new ways for representing and manipulating sound. As the prototype of a reproduction apparatus, the tape machine was able to radically transform and manipulate recorded sound despite the fact that electromagnetic tape symbolizes linear thinking. On the other hand, digital systems of audio recording introduced non-linear representation in which sound is broken down into an atomic dot-like structures that disintegrate into a mosaic of numbers as the bond with temporal sound tissue dissolves. The fragmented granular structure of the sound, which can be manipulated by computers and artificial intelligences, replaces linear thinking and promotes a consciousness of the microstructure of any given sound.

As Pousseur observed, electroacoustic music articulates a continuous interaction between different levels of sound organization, so that it becomes difficult "to draw a precise boundary between internal composition of sound and higher levels of composition" [4, 82]. A myriad of sound poetics emerged within the electroacoustic paradigm such as soundscape composition, deep listening, live-electronics, and other musical distinctions involving vocal, instrumental, or electronic sounds. The electroacoustic paradigm extended sound perception and consciousness, especially in the way it relates to microscopic and macroscopic levels of sonic composition. The opposition of macro/micro sound, along with the methodic use of music apparatuses, is a signature of the electroacoustic paradigm symbolizing a desire for intensification of the living experience.

2 Sound Embodiment and Sound Space

Human *embodiment* can be seen as a mediator between technology and the world. In traditional acoustic music, gestures are made distinctive through specific features such as articulations, dynamics, timing, rhythm, meter, texture, and timbre. In electroacoustic music, the body's gestural interface – visual, acoustic, and tactile – facilitates new kinds of interactive and intersubjective communication. For both acoustic and electroacoustic music, gesture articulates not only the perception of nuance, cognition, and affect — but also negotiates the understanding of higher sound and musical structuring through internal synthesis and integration of elements.

Embodiment and *gestural* activity emerge as key concepts in discussions of space in electroacoustic music [5]. The increasing focus on the multiple connections between

sound, body, and listening reaffirms the notion of space as enacted experience. This represents a significant shift from the typological and morphological approaches of sound to new formulations based on more synthetic, phenomenological, and ecological categories. Nevertheless, a problem persists in the theoretical and analytical discussion, namely the distinction between "internal" and "external" sound space. The structural coupling of internal and external references, as pointed out by Luhmann [6] in the realm of his autopoietic theory of social systems, poses the question: How do artistic objects articulate and combine perception and communication?

In Luhmann's response, sound space must be defined not in terms of sonic qualities, but as a *mode of operation of consciousness that gives form to the perception of space within the acoustic environment*. Similar to the operation that produces polyphony, sound space is the form of the difference between *self-reference* (internal world) and *hetero-reference* (external world) in acoustic perception. This definition implies that consciousness has to establish the boundaries that connect and disconnect the perception of sound phenomena to the perception of space. The definition of sound space is a particular embodiment based on the possibility of perceiving sounds as meaningful elements.

In opposition to instrumental and vocal sounds that are *tightly coupled* with the body and the objects that produced them, electroacoustic sounds can be seen as *loosely coupled* because they leave room for multiple combinations. The sound recording of a voice, instrument, or environment is an inscription and re-creation of sound waves that can be transformed in different ways and turned into something completely altered from the original sound. Luhmann introduced the opposition between *loose coupling* and *tight coupling* to account for the difference between media and form. Media is a loose coupling of elements, something more abstract and fluid — while form is a tight coupling of elements, something more stable and tangible. [6, 102-132]. Electroacoustic music is a disembodied entity as sound frees itself from the body. Therefore, electroacoustic composition requires a process of re-actualization of meaning in order to endow sounds with a bodily, spatial memory.

From the beginning, space has been a functional and operational category of electroacoustic composition. Sound space composition then became more fully realized with the introduction of multi-channel audio technology. Prototypes of multi-channel technology consist of the four-track tape recorder and the quadraphonic speaker system surrounding the listener: a stereo pair in the front and another in the back. Through the use of this technology in the late 1950s and 1960s, composers began to create pieces in which sounds were designed for specific positions in space. Once space became a parameter of composition, sound developed a "tactile" dimension. Similar to a body, it occupies a unique position in the space from which it can exclude other spaces.

3 Audiovisual Composition

Currently, the concept of audiovisual art is framed by the dominant role film and television play in our society, founded on technology of sound and image reproduction invented in the second half of the 19th Century. Cinematography, as an audiovisual art

that emerged from the movement of technical images, elevated film to the most popular artistic form in human history. With the supremacy of the moving image, especially during the silent film era, it was possible to cross borders and establish patterns of transnational communication. As the sound film quickly prevailed as a product of mass consummation, cinema, and later television, shaped the perception of sound and image until the end of the 1980s when digital technology set the stage for radical transformation. The popularization of personal computers, mobile devices, and networks of information and communication, began to reframe the creativity of audiovisual art. Technology propelled convergences of sound, image, space, and performance to create new architectures of collaboration giving rise to new kinds of transnational dialogues. As the traditional structures of creation and production of audiovisual art underwent this enormous change, new artistic forms of audiovisual composition began to emerge.

In the universe of electronic music, there has been a growing interest in audiovisual composition with more electroacoustic works being coupled with video, and mixed works combining electroacoustic sounds, live performance, and visual projection. Audiovisual composition has the potential to bring electroacoustic music to a broader audience, as it addresses a multimodal perception and sensibility. It reveals two important components: the convergence of fields and perceptions as well as the creation of a diversity and differentiation of forms. Composers of audiovisual works have much to consider. Based on their initial motivation to create a new piece, they are faced with the question of which will be more important – the music, the visuals, or the combination of the two? They must consider how the sound and image relate to each other as they attempt to intensify the immersive, sensorial experience and try to raise the consciousness of the interconnection between hearing and listening as a mode of being in the world. If they fail to achieve these objectives, should the audiovisual composition be considered just another distraction reinforcing the patterns of entertainment and diversion? As a society inhabited with myriad trivial objects and gadgets of audiovisual technology taking a hold on our existence, we have become saturated by the torrent of audiovisual impressions. Faced with this flood of information that can lead to a state of entropy, it is important to develop a critical reflection on audiovisual communication. We need a comprehensive account of the relation between sound and image beyond the conventional form of cinema in order to understand its full creative potential. It is necessary to deconstruct the hegemonic discourses and point out the broad spectrum of possibilities and diversity of forms within audiovisual composition.

4 The Electronic Music Video

The music video, which emerged in tradition of electronic music, is a contemporary form of audiovisual composition coupling electronic sounds with image projection that enjoys growing interest and is developing into a sub-genre of electroacoustic music. The music can be “heard” and “seen” at the same time. The audiovisual merge seems to have the potential to make the music more accessible to a broader audience. But here one has to raise the following question: Does the multimedia intensify the sensorial experience and make it thus more attractive, or does it simply provide a distraction that

reinforces the patterns of entertainment and diversion of the consume society? Whatever the answer may be, embedding the music into an audiovisual form provides the listener with an immersive experience that is functionally linked to the situation of the movie theater: the music is projected into the room through loudspeakers, the sound surrounds the bodies, while the image projected onto a screen—usually located in front of the audience—focuses the audience's attention on an illuminated surface.

The electroacoustic music video relating sound, image, and space is primarily an immersive experience that can be also integrated with other forms, such as the concert with live music performance (vocal, instrumental, and/or electroacoustic music); the performance with dance, acting, etc.; the installation; and so on. Traditionally, the audiovisual art is structurally coupled with the space both as physical and social medium. The immersive experience relates physical presence to social presence. By contrast, watching an electroacoustic music video on a computer, on internet, or on a mobile device is mainly an individual experience, in which the embodied experience is dispersed along a spectrum of possibilities emerging from the interaction with the technological environment.

As an illustration of audiovisual composition, we would like to examine the piece *Vega_S* (2019)[2] by the distinguished Russian composer Igor Kefalidis (b. 1941). Kefalidis' profound interest in electroacoustic music has resulted in a long period of composing pieces exclusively with electroacoustic sounds — most in combination with solo instruments, chamber music, and orchestra. His creativity reaches into the fields of dance and audiovisual composition and the relationship between sound and image plays a crucial role in his recent work, in which he has been collaborating with visual artists. Most recently, he has been adopting new tools to create synthetical images.

Vega_S (2019) – length 13'05" – for electronic sounds and video is a remarkable piece that represents a mature stage of Kefalidis' audiovisual composition style. Here, the electroacoustic music seems to bring forth the imagery, as though the sounds are endowed with visual symbolism. The visual composition by Andrew Quinn takes advantage of the imaginative character of the music and seeks to create an organic relationship through the use of a thin white vertical line in the middle of the screen that varies in brightness according to the music. The line turns into a narrow dark space separating two walls that constitute the main element of the visual composition. The walls are curved with a translucent and pixelated structure in black and white that continuously rotate in opposite directions, changing speed according to the sonic variations of the music. Figures appear and disappear in the narrow space between the walls and the spaces and on their left and right sides and the pulsing activity of these intermittent elements are in sync with the music. At 5'20", a strong beat punctuates the visual composition and the music speeds up and ascends in a pseudo–quotation of a short compelling rock guitar solo (8'35" – 8'48"). As the musical energy increases, colorful strips are introduced in the wall landscape, rotating ever faster and disrupting the visual symmetry to create a fragmented, fast-moving kaleidoscopic image to accompany the rock guitar. The electronic music creates the impression of fluid space as the sound objects and events seem to move closer and then farther away. The visual composition explores

[2] Available at <https://youtu.be/QLGKroHIpaA> (accessed June 1, 2021).

the fluidity of the space by creating a kind of futuristic landscape that constantly moves without a clear direction. The audiovisual composition presents us with the ambiguity of experiencing a calculated universe while simultaneously allowing us the chance to move between the world of algorithms. Overall, it infuses us with energy and hope as it suggests the need to disrupt hegemonic structures of power to escape via beams of flight leading to unknown territories. *Vega_S* is an accomplished example of the synergy of sound and image. The multiplicity of connections between electroacoustic sounds and synthetic images portray the massive potential of audiovisual composition.

5 From Soundscape to Telematic Immersion

Telematic music is an attempt to make a synthesis of two different types of communication: (1) The communication of chamber music, which occurs in the physical medium with bodies producing gestures that are translated into sounds; (2) the communication of electronic music, which occurs in the virtual medium with apparatuses producing programs that are translated into sounds or images. Unlike traditional chamber music, which is structured as a succession of linear events such as themes and variations, telematic music creates a dialog that "occurs in simultaneous time and space, and all players in all places make decisions relating to themes and their variations all at once" [7]. Telematic music offers the possibility to reshape musical performance in virtual spaces by reconstructing the subjectivity with the *experience of presence*.

As an illustration of telematic music, we will discuss *Mojave* (2021)[3] – length 8:53" – a collaborative work for flute, electronics and video that unfolds an aesthetics of audiovisual immersion with telematic performance. The work was developed on the basis of 3D video and ambisonics audio recordings on the desert of Mojave (California) in January 2019. Cassia Carrascoza created a performance for this specific site physically interacting with the landscape and improvising with sounds exploring extended techniques for flute and bass flute. Paulo C. Chagas composed a score for flute and live electronics exploring algorithms of delay and feedback, which create a universe oscillating between latencies and synchronies. Different versions of the piece were created for audiovisual media and live telematic performance. Mojave is a multilayered audiovisual composition that reflects on the presence and absence as vectorial forces of creativity. The contrast between the vast desert landscape and the confined telematic environment evokes the existential feelings of eternal and transitory, the finite and the infinite, and the anxiety we current experience between isolation and the opportunity to immerse ourselves into virtual worlds.

Conceptually, *Mojave* is part of the large-scale research project *Sound Imaginations*, which aims to investigate listening cultures and different categories of listening.[4] The emblematic notion of *soundscape* proposed by the Canadian composer and scholar Murray Schafer in the 1970s [8] is a key concept for observing the sonic environment, which includes not only the "natural" sounds but also the entire culture that

[3] Available at <https://youtu.be/GB-KwDOImho> (accessed June 01, 2021).

[4] *Sound Imaginations* (2020) immersive surround sound and 3D video installation available at <https://ucrarts.ucr.edu/Exhibition/sound-imaginations> (accessed June 01, 2021).

characterizes the sonic environment of any specific space or object of study. Driven by Schafer's ideas, many scholars and artists have been pursuing the mapping of historical and contemporary soundscapes and observing the transformation of soundscapes in the industrial and digital societies. Many authors have criticized Schafer for having projected the problematic concept of "soundscape" borrowed from visual art into sound studies as it suggests a static perspective rather than the moving and surrounding characteristic of sound phenomena. Also, it implies a division between hearing and seeing, which is highly problematic in the contemporary world shaped by the connective reality of audiovisual and multimedia technology.

Feld [9], for instance, proposes the concept of acoustemology – the union of acoustics and epistemology – that investigates the primacy of sound as a modality of knowing and being in the world. Soundscapes are not just physical exteriors, they are perceived and interpreted by human actors and are invested with significance by those whose bodies and lives resonate with them in social time and space. As a cultural system, sound both emanate from and penetrates bodies; hearing and producing sound are thus embodied with competencies that situate actors and their agency in particular historical worlds.

The compositional concept of *Mojave* was elaborated on the basis of the semiotic square proposed by Hayles [10] that reconstructs the distinction between *randomness* and *pattern* in the so-called *posthuman* society while emphasizing the role of *embodiment* and *materiality* in the processes of constituting meaning. Hayles' semiotic square (Figure 1) has two axes: the main axis is the distinction between *presence* and *absence*; the secondary axis is the distinction between *randomness* and *pattern*. Two diagonals that connect these two axes trigger a dynamics of signification. The diagonal connecting presence and pattern conveys *replication*; the diagonal connecting absence and randomness signals *disruption*. The interplay between presence and absence shapes materiality; the interplay between randomness and pattern gives rise to information [10, 247-251].

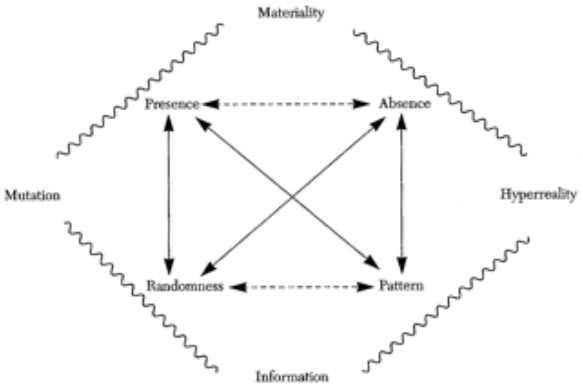


Fig. 1: Hayles' semiotic square of the posthuman society [10, 249].

On the site of the desert of Mojave, Cássia Carrascoza developed a performance with flute and bass flute that articulates a dialectics of presence/absence emerging from the

auditory and visual perception of the soundscape/landscape. For instance, the presence the strong wind blowing through the vast space of the desert – which also autonomously activated flute sounds –, and the sounds produced by the crackling of small stones as one moves across the uneven desert ground, these are two elements that were integrated in the performance, along with long sounds and extended flute techniques. The 3D images move around Cássia as focal point, a central figure that captures the human presence in the emptiness of the desert landscape, which symbolizes void and absence. Starting from this focal point, the movements unfold edges, diagonals, curves, rotations, and circular movements that opens up a constant play of spiracle shapes, a vortex of 3D images that pushes things beyond the center, creating a path of decentering moving along both the axis of presence/absence and randomness/pattern. The musical composition associates visual imagery with the spherical sound perception of the ambisonics technology. It explores a vocabulary of sound shapes and colors, sound objects, events or movements that tease out the decentering of the listener, which is sometimes synchronized and sometimes out of sync with the visual.

Mojave is a collaborative work between a composer and a performer acting as equal partners that takes into account the new fields of creativity emerging through the convergence of sound, image, and the development of new architectures of collaboration. It addresses resources, approaches, and strategies of audiovisual composition in an environment where information is embodied in complex heterogenic and polyphonic structures of subjectivity. The piece exists in different versions including a real-time telematic performance.[5] As pointed out by Guattari [11] [12], subjectivity is no longer restricted to human consciousness, but incorporates the body of technology through what he defines as "machinic assemblages". Creativity no longer depends on personal identity and subjectivity but on the particular assemblage that happens in connection with technological bodies that extend the framework of cognition and meaning. The structure of the "machinic assemblages" can be defined as "polyphonic, as it articulates a multiplicity of human and non-human subjects bringing several simultaneous and independent levels of perception and meaning" [1, 106].

6 Conclusion: Telematic Embrace

As Heidegger [13] argues, modern technology has changed our sense of the world as it tends to reduce everything into mere resources, including human beings. The programmatic magic of technical apparatuses, including artistic apparatuses that produce synthetic sounds and images, tends to eliminate critical thinking, replacing historical consciousness with a second-order magical consciousness that reduces culture to its lowest denominator. With the technical apparatus, relations of power move from physical objects to a symbolic level of programs and operators.

The telematic paradigm embraces the communicative complexity that emerges from the convergence of telecommunications and information processing in today's society. Flusser [7] believes that telematic communication has the potential to radically

[5] Available at <https://youtu.be/onuWdf92KrI> (accessed June 1, 2021).

transform the way we communicate. Telematics can reverse the natural tendency of entropy – the state of randomness in which information is unpredictable and therefore impossible – by converting historical and discursive thinking into dialog. In Flusser's telematic dialog, man and apparatuses act as partners devoting themselves to the systematic generation of information through a playful game. The telematic dialog embodies Flusser's utopia of freedom as a struggle against entropy, which emancipates man from the controlling functionality of the machine.

The possibilities of artistic collaborations between participants in remote locations, interacting via electronic networks, can facilitate interactive art and interdisciplinary, as Ascott pointed out in his seminal writing of 1960s [14, 109-156] The telematic paradigm involves not only the technology of interaction among human beings but between the human mind and artificial systems of intelligence and perception. It transcends the body, amplifies the mind into unpredictable configurations of thought and creativity, and can contribute to the emergence of a global consciousness. The holistic potentiality of telematic art supports Ascott's metaphor of love in the telematic embrace. Like gravity, passionate attraction draws together human beings an connects then. Global telematic embrace would constitute an "infrastructure for spiritual interchange that could lead to the harmonization and creative development of the whole planet" [14, 245].

References

1. Chagas, P. C.: Unsayable Music: Six Essays on Musical Semiotics, Electroacoustic and Digital Music. Leuven University Press, Leuven (2014).
2. Husserl, E.: Zur Phänomenologie des inneren Zeitbewußtseins. M. Nijhoff, The Hague (1966).
3. McLuhan, M.: Understanding Media: The Extensions of Man. The MIT Press, Cambridge (1994).
4. Pousseur, H.: Fragments théoriques I sur la musique expérimentale. Editions de l'Institut de Sociologie de l'Université Libre de Bruxelles, Brussels (1970).
5. Smalley, D.: Space-form and the Acousmatic Image. Organized Sound 12(1), 35-58 (2007).
6. Luhmann, N.: Art as Social Systems. Stanford University Press, Stanford (2000).
7. Flusser, V.: Into the Universe of Technical Images. University of Minnesota Press, Minneapolis (2011).
8. Schafer, M.: The Soundscape: Our Sonic Environment and the Tuning of the World. Destiny Books, Rochester (1994).
9. Feld, S.: A Rainforest Acoustemology. In: The Auditory Culture Reader, pp. 223-279. Bull, M. and Back, L (eds.), Berg, Oxford; New York (2003).
10. Hayles, K.: How We Became Posthuman: Virtual Bodies in Cybernetics, Literature, and Informatics. University of Chicago Press, Chicago (1999).
11. Guattari, F.: Chaosmosis. Galilée, Paris (1992).
12. Guattari, F.: Machinic Heterogenesis. In: Rethinking Technologies, pp. 13-17. University of Minnesota Press, Minneapolis (1993).
13. Heidegger, M.: The Origin of the Work of Art. In: Basic Writings, pp. 139-212. Harper Perennial, New York (2008).
14. Ascott, R.: Telematic Embrace: Visionary Theories of Art, Technology, and Consciousness. University of California Press: Berkeley (2003).

CROCUS: Dataset of Musical Performance Critiques
Relationship between Critique Content and Its Utility

Masaki Matsubara[1*], Rina Kagawa[1*], Takeshi Hirano[2], and Isao Tsuji[3]

[1] University of Tsukuba
[2] University of Electro-Communications
[3] Senzoku Gakuen College of Music, Kunitachi College of Music
masaki@slis.tsukuba.ac.jp

Abstract. In musical performance education, verbal as well as non-verbal information is used to convey knowledge. In the current social situation, the demand for remote and asynchronous lesson is increasing, and it is not clear what types of verbal information should be used. In this study, we collected 239 critique documents written in Japanese by 12 teachers for 90 performances of the same 10 orchestra studies of the oboe by 9 students. We categorized the critiques and found that their content differed more by the teacher than by the piece or the student. We also found that the category of *giving a practice strategy* was particularly valued by students.

Keywords: Database, Music Education, Verbal Information

1 Introduction

Playing musical instruments has traditionally been taught in-person and was considered unsuitable for virtual learning environments. However, the COVID-19 pandemic has led to increased demand for online music tuition [1, 17]. In musical performance tuition, knowledge is conveyed using both non-verbal information such as singing melodies and gesturing, and verbal information such as pointing out mistakes [7, 12, 22]. Verbal information is essential for conveying how the learner's performance sounds, why they did not play well, and how they should practice. An advantage of online music tuition is that space and time do not necessarily have to be shared, thus allowing for remote and asynchronous teaching. However, due to the low resolution of online video/audio communication, there is a limitation in the use of non-verbal information, as it is difficult to convey complex body movements and high-quality sound performances. Therefore, the importance of verbal information in music tuition, especially in the critique documents of asynchronous online performance education, is expected to increase [10].

However, teaching using words is not easy. In our preliminary survey of nine music college students and one hundred people with musical performance experience, they reported a good impression of their musical experience, although some were not satisfied with their teacher's instructions. We collected free-text responses about

[*] Two authors equally contributed to this research.

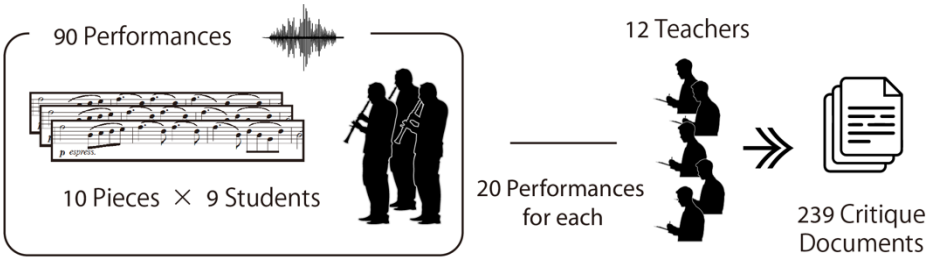


Fig. 1. Overview of the construction of the musical performance critique dataset.

dissatisfaction with the instruction and categorized the results as the pertaining to three issues: (1) content of performance instruction (e.g., "I would have preferred instruction based on facts," "lack of concrete advice"), (2) consistency of instruction over multiple lessons (e.g., "completely different or inconsistent attention from lesson to lesson"), and (3) wording of instruction not related to performance (e.g., "all the teacher did was scold without much praise").

We believe that one reasons for these problems is the lack of teaching protocols in performance instruction and the lack of systematic clarification of what should be verbalized to benefit learners. At present, however, the empirical knowledge of what types of instruction are provided is not widely shared, even among students who aspire to become professionals.

This paper introduces an open dataset of musical performance critiques in Japanese, called *CROCUS* (CRitique dOCUmentS of musical performance), to promote music education through the study of verbal information in performance instruction. We define critique documents as comments written by teachers to give feedback on a performance. We collected 239 critique documents from 12 teachers for 90 performances of 10 pieces by 9 students (Fig. 1). Because music college classes are conducted online at present, we collected recordings and critique documents similar in manner to those in asynchronous classes. This dataset allows us to compare critiques for each piece, student, and teacher. We examined which critique contents were perceived by the performers as useful instruction. Specifically, we analyzed types of verbal information, measured the perceived utility of critiques, and examined differences in utility scores among teachers, students, and pieces.

The contributions of this paper are as follows:

- We constructed an open dataset of 239 critique documents of 90 musical performances of 10 oboe orchestral studies[1].
- We quantitatively demonstrated that the content of the critique documents varies more by teacher than by piece or student.
- We collected evaluations of the critique documents from people with musical experience and examined the types of verbal information to determine what in the critique documents was described as having high utility.

[1] Dataset is public on https://doi.org/10.5281/zenodo.4748243

Table 1. List of Pieces

ID	Composer	Piece
p01	L. v. Beethoven	Symphony No. 3 in E flat Major 'Eroica', Op. 55
p02	G. A. Rossini	'La Scala di seta' Overture
p03	F. Schubert	Symphony No. 8 in B Minor D.759 'Unfinished'
p04	J. Brahms	Violin Concerto in D Major, Op. 77
p05	P. I. Tchaikovsky	Symphony No. 4 in F minor, Op. 36
p06	P. I. Tchaikovsky	"Swan Lake", Ballet Suite, Op.20a
p07	N. Rimsky-Korsakov	"Scheherazade", Symphonic Suite, Op. 35
p08	R. Strauss	"Don Juan", Symphonic Poem, Op. 20
p09	M. Ravel	Le Tombeau de Couperin I.Prelude
p10	S. Prokofiev	"Peter and the Wolf", Symphonic Tale, Op. 67

2 Related Work

2.1 Music Database for Research

Numerous music databases have been published, with, for example, performance recordings data [14], metadata (genre, composer, lyrics, etc. [15, 28, 32]), musical scores (MIDI [20], piano notation [11]), information associated with scores (fingering [25], music analysis [16]), and other multimodal information [23]. There are also databases about the human aspects involved in music, including emotions [5], listening history [27], and performer interpretations [18, 19, 26], but, to the best of our knowledge, no database shares critiques in performance education.

2.2 Teaching Behavior in Musical Performance Tuition

Teaching behavior in musical performance tuition has been studied in the music education field, including comparison of teacher levels [13], analysis on time allocation [4], comparison [33] and categorization [29, 30] of verbal and non-verbal information, and teacher‐student interaction [9]. These studies targeted the transcription of speech in interactive instruction. Our study focused on critique documents that can be used for asynchronous education.

Regarding the utility of instruction, one study compared verbal and non-verbal instruction [6], and another summarized the evaluation of its usefulness [8]. These studies were based on five or fewer performances. We conducted a large-scale study and clarified the relationship between the verbal information and its utility.

3 Method

We constructed the CROCUS dataset by collecting performance recordings and critique documents. Then, all comments in the documents were annotated. Finally, the perceived utility of every document was evaluated. All collection procedures were approved by the ethical review boards of the University of Tsukuba, Senzoku Gakuen College of Music, and Kunitachi College of Music.

Table 2. Types of verbal information in this study

	Type	Definition
Adopted and adapted from Carlin [3]; Zhukov [34]; Simones [30]	Giving Subjective Information (GSI)[2]	Providing general and/or specific conceptual information based on the teacher's subjectivity.
	Giving Objective Information (GOI)[2]	Providing general and/or specific conceptual information based on objectively referable events or concepts.
	Asking Question (AQ)	Enquiring.
	Giving Feedback (GF)	Evaluation of a student's applied and/or conceptual knowledge.
	Giving Practice (GP) Strategy	Providing suggestions for ways to practice a particular passage or discussing a practice schedule.
	Giving Advice (GA)	Giving a specific opinion or recommendation to guide the student's action toward the achievement of specific musical aims, without demonstration or modeling.

3.1 Constructing the CROCUS Dataset

A total of 90 performances (10 pieces by 9 music college students majoring in the oboe) were recorded. As online lessons have become the norm in the music colleges due to COVID-19, we adopted a comparable situation. Each student played in a less reverberant and less noisy environment at home, about 1 m away from the recording device (Roland R-07). Tuning and recording level were adjusted at the beginning of the recording. We selected the 10 pieces in Table 1 to balance difficulty, style, form, and era.

3.2 Annotating Types of Commentary in CROCUS

We adopted and adapted Simones' definitions [30], as shown in the Table 2[3]. One of these six types was annotated to each sentence. Sentence breaks were periods or exclamation marks. When a sentence was judged as consisting of multiple types, they were separated by a comma. Two annotators annotated all 239 documents. If the annotations did not match, the final annotation was decided through discussion. The Cohen's kappa coefficient was 0.96.

3.3 Evaluating Perceived Utility of Critique Documents

The perceived utility of the collected critiques was examined by 200 people who answered the question "Do you think that this document is useful for future performances?" by using an 11-point Lickert-like scale (10: useful – 0: useless). Participants responded to 25 randomly selected critique documents. This question is referred to as Q1.

[2] Originally "Giving Information." Divided by the authors.

[3] Types of "Demonstrating", "Modelling", and "Listening/Observing" were omitted because these actions are not observed in a written text.

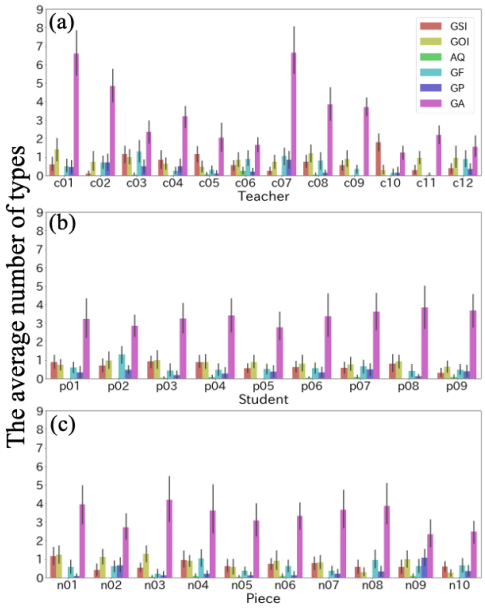


Fig. 2. Number of commentary types for each document per (a) teacher, (b) student, and (c) piece. The error bar indicates the standard deviation.

Detailed Analysis of the Utility: Utility is not limited to usefulness for future performances. Therefore, we referred to the usefulness perspective used in software requirement specifications [21] and accounting documents [31] for eight other items. The questions we used were as follows. All questions were asked in the form of "Do you think that this document —?" Q2: is readable, Q3: is understandable, Q4: has language not related to future performances, Q5: is not ambiguous, Q6: contains only statements related to the future performances, Q7: is consistent, Q8: can be verified by listening to the performance, and Q9: allows you to refer to the relevant part in the score from the content described.

4 Results

4.1 Constructing the CROCUS dataset

A total of 239 critique documents[4] were provided by 12 teachers who are currently with or formerly belonged to well-known music colleges, orchestras, and brass bands in Japan. Each teacher wrote critique documents assuming the usual lessons for a total of 20 performance recordings. The 20 performances were selected in a counterbalanced manner with the following constraints: each piece was reviewed by at least two teachers, and every teacher reviewed at least one performance for every student. Due to the current social situation, the critique documents were also written at the teacher's home.

[4] One critique was lost during the collection process.

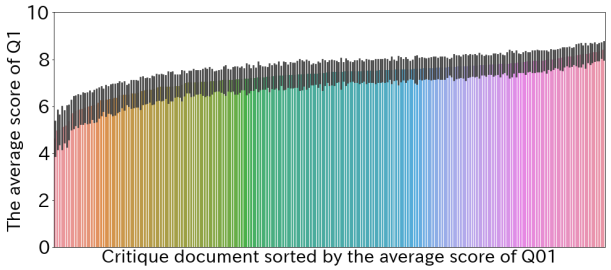


Fig. 3. Average score of Q1 for each critique document (sorted by Q1 score).

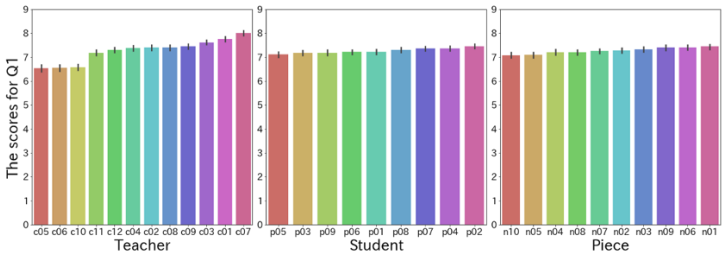


Fig. 4. Average scores for Q1 by teacher, student, and piece (sorted by Q1 score).

An example of a critique document is as follows:

The difficult passages are performed well here. If I were to ask for more, the sound is almost "too" fulfilling — it feels like a pancake with slightly too much syrup on. That may not be the best comparison...

4.2 Annotating Types of Commentary in CROCUS

GSI, GOI, AQ, GF, GP, and GA appeared in 47.28%, 54.81%, 3.34%, 39.33%, 22.18%, and 93.72% of the documents, respectively. The average (and standard deviation) of the number for each category per document was, 0.70 (0.90), 0.85 (1.00), 0.03 (0.18), 0.61 (0.88), 0.33 (0.70), and 3.33 (2.50), respectively. Fig. 2 shows that the differences in the content of documents were larger among teachers than among songs or students.

4.3 Evaluating Perceived Utility of Critique Documents

Our results showed that the critique documents had a variety of utility scores, and there were documents that the readers perceived as less useful (Fig. 3). Since the null hypothesis that the distributions of Q1 values for each teacher, student, and piece (Fig. 4) were normal was rejected by the Shapiro-Wilk test, the Kruskal-Wallis test was used. The null hypothesis that the Q1 values that the reader perceived useful were equal among all teachers and the null hypothesis that they were equal among all pieces were rejected ($p \leq 0.001$, the effect size was small).

Table 3. The p-value and effect size of the Kruskal–Wallis test conducted for each question item for each teacher, student, and piece.

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Teacher	***(s)	***(m)	***(s)	***(s)	***(m)	***(m)	***(s)	***(s)	***(m)
Student	**(vs)	***(vs)	***(vs)	***(vs)	***(vs)	***(vs)	*(vs)	***(vs)	***(vs)
Piece	***(vs)	***(vs)	− (vs)	− (vs)	***(vs)	***(vs)	*(vs)	***(vs)	***(s)

* $p \leq 0.05$, ** $p \leq 0.01$, *** $p \leq 0.001$; m, s, and vs mean moderate, small, and very small, respectively.

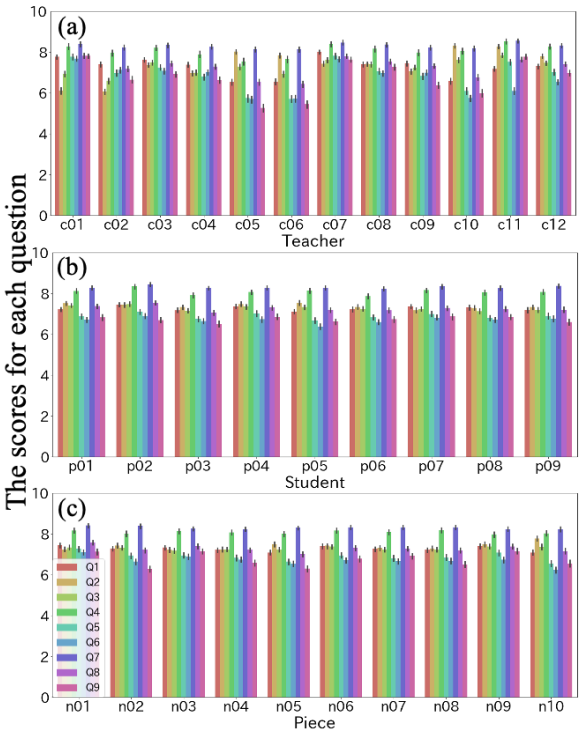


Fig. 5. Question scores for each document per teacher, student, and piece.

Detailed Analysis of the Utility: Table 3 presents the results of the Kruskal–Wallis test performed for the average score of each question (Fig. 5). The difference between the teachers was more remarkable than that between the students or the pieces for all questions. This result was consistent with previous research showing that the usage of words differs depending on the teachers [22]. The differences among the teachers were particularly large regarding whether the commentary was easy to understand (Q3), not ambiguous (Q5), contained only descriptions related to future performances (Q6), and can refer to the relevant part in the score (Q9). Detailed statistical analysis of the utility score of critiques is described in [24].

The critique documents with the highest average Q1 and that with the lowest average value are shown as follows (types are annotated with square bracket):

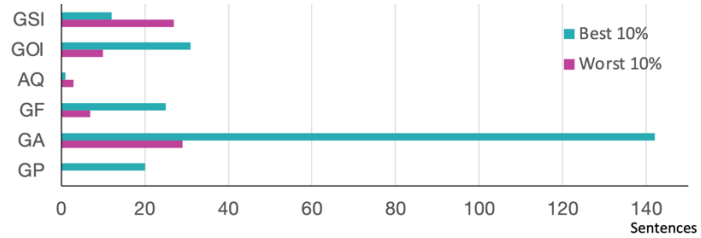


Fig. 6. Histogram of types in best 10% and worst 10% of critique documents.

The highest-rated critique (n09-p04-c03) (Q1: 8.41 ± 1.44)
[GSI] *I feel that this performance is very good, and it leaves a very favorable impression.* [GA] *Because of this, I would like you to be a little more careful regarding the nuances of the performance.* [GP] *Please practice the grace notes in bars 2 and 4 again by themselves. The same for bar 10.* [GF] *There is always a mistake in the E♭-H transition in bar 11.* [AQ] *Perhaps it is a problem with the tuning of the instrument?* [GP] *Please perform this part slowly and check carefully.* [GF] *If it is not a tuning problem, then I believe it is a fingering or breathing problem.* [GP] *Please practice carefully and check if the breathing and fingering are both coordinated properly.* [GA] *In the second half, there is tenuto on the high E and D notes.* [GA] *Please endeavor to perform each note carefully with nuance.*

The lowest-rated critique (n04-p06-c05) (Q1: 4.63 ± 2.61)
[GSI] *The melodies are performed beautifully and vibrantly, almost as if I could hear an orchestra performing.* [GSI] *The phrasings are well expressed for the piece, and it was lovely.*

5 Discussion

As Fig. 3, Fig. 4, and Table 3 indicate, the utility scores of critiques differed more greatly than by piece or each student. Teachers whose comments received high utility scores (e.g., c07, c01, and c03) provided information on the performance based on objective evidence (GOI), indicated the direction that the student should aim for (GA), and suggested practice strategies (GP). Fig. 6 is a histogram of types in the best 10% and worst 10% critique documents in terms of Q1 score. Giving practice strategies (GP) were observed in the best 10% and not in the worst 10%. Giving advice (GA), providing information based on objective facts (GOI), and giving feedback (GF) were also more common in highest-rated critiques.

6 Conclusion

We published the CROCUS dataset as a starting point for investigating the use of language in critique documents. The dataset clarified that the contents of critiques varied most by teacher, and suggested that the category of giving a practice strategy was valued by students.

Since the dataset was constructed at an early stage of the project, the instrument was limited to the oboe. Whether the findings are generalizable remains as an open question. We would like to explore more instruments and discuss whether a good critique document structure has a common characteristic. The student participants were limited to music college students; thus, we would like to explore the topic at various levels of experience, such as professional and amateur students. Finally, the study was conducted using only Japanese. In the future, it will be necessary to conduct comparisons among multiple languages and discuss differences between languages and cultures [2].

Acknowledgments

This study was partially supported by JST-Mirai Program Grant Number JPMJMI19G8, and JSPS KAKENHI Grant Number JP19K19347. We would like to thank all the performers and teachers who participated in the data collection. We would also thank to those who helped us with data annotation and evaluation.

References

1. Bayley, J. G. and Waldron, J.: "It's never too late": Adult students and music learning in one online and offline convergent community music school, Int. J. Music. Educ., Vol. 38, No. 1, pp. 36–51 (2020).
2. Campbell, P. S.: Lessons from the world: A cross-cultural guide to music teaching and learning, MacMillan Publishing Company (1991).
3. Carlin, K. D.: Piano pedagogue perception of teaching effectiveness in the preadolescent elementary level applied piano lesson as a function of teacher behavior, PhD Thesis, Indiana University (1997).
4. Cavitt, M. E.: A descriptive analysis of error correction in instrumental music rehearsals, J. Res. Music. Educ., Vol. 51, No. 3, pp. 218–230 (2003).
5. Chen, Y.-A., Yang, Y.-H., Wang, J.-C. and Chen, H.: The AMG1608 dataset for music emotion recognition, ICASSP, pp. 693–697 (2015).
6. Dickey, M. R.: A comparison of verbal instruction and nonverbal teacher-student modeling in instrumental ensembles, J. Res. Music. Educ., Vol. 39, No. 2, pp. 132–142 (1991).
7. Dorman, P. E.: A review of research on observational systems in the analysis of music teaching, Bull. Counc. Res. Music. Educ., pp. 35–44 (1978).
8. Duke, R. A.: Measures of instructional effectiveness in music research, Bull. Counc. Res. Music. Educ., pp. 1–48 (1999).
9. Duke, R. A. and Simmons, A. L.: The nature of expertise: Narrative descriptions of 19 common elements observed in the lessons of three renowned artist-teachers, Bull. Counc. Res. Music. Educ., pp. 7–19 (2006).
10. Dye, K.: Student and instructor behaviors in online music lessons: An exploratory study, Int. J. Music. Educ., Vol. 34, No. 2, pp. 161–170 (2016).
11. Foscarin, F., McLeod, A., Rigaux, P., Jacquemard, F. and Sakai, M.: ASAP: a dataset of aligned scores and performances for piano transcription, ISMIR, pp. 534–541 (2020).
12. Froehlich, H.: Measurement dependability in the systematic observation of music instruction: A review, some questions, and possibilities for a (new?) approach., Psychomusicology, Vol. 14, No. 1-2, p. 182 (1995).

13. Goolsby, T. W.: Verbal instruction in instrumental rehearsals: A comparison of three career levels and preservice teachers, J. Res. Music. Educ., Vol. 45, No. 1, pp. 21–40 (1997).
14. Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R.: RWC Music Database: Popular, Classical and Jazz Music Databases., ISMIR, pp. 287–288 (2002).
15. Goto, M., Hashiguchi, H., Nishimura, T. and Oka, R.: RWC Music Database: Music genre database and musical instrument sound database, ISMIR, pp. 229–230 (2003).
16. Hamanaka, M., Hirata, K. and Tojo, S.: GTTM database and manual time-span tree generation tool, SMC, pp. 462–467 (2018).
17. Hash, P. M.: Remote learning in school bands during the COVID-19 shutdown, J. Res. Music. Educ., Vol. 68, No. 4, pp. 381–397 (2021).
18. Hashida, M., Matsui, T. and Katayose, H.: A New Music Database Describing Deviation Information of Performance Expressions., ISMIR, pp. 489–494 (2008).
19. Hashida, M., Nakamura, E. and Katayose, H.: Constructing PEDB 2nd Edition: a music performance database with phrase information, SMC, pp. 359–364 (2017).
20. Hawthorne, C., Stasyuk, A., Roberts, A., Simon, I., Huang, C.-Z. A., Dieleman, S., Elsen, E., Engel, J. and Eck, D.: Enabling Factorized Piano Music Modeling and Generation with the MAESTRO Dataset, ICLR (2019).
21. IEEE: Recommended Practice for Software Requirements Specifications, IEEE Std 830-1998, pp. 1–40 (1998).
22. Lehmann, A. C., Sloboda, J. A., Woody, R. H., Woody, R. H. et al.: Psychology for musicians: Understanding and acquiring the skills, Oxford University Press (2007).
23. Li, B., Liu, X., Dinesh, K., Duan, Z. and Sharma, G.: Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications, IEEE Tran. Multimedia, Vol. 21, No. 2, pp. 522–535 (2018).
24. Matsubara, M., Kagawa, R., Hirano, T. and Tsuji, I.: Analysis of Usefulness of Critique Documents on Musical Performance: Toward better Instructional Document Format, ICADL (to appear) (2021).
25. Nakamura, E., Saito, Y. and Yoshii, K.: Statistical learning and estimation of piano fingering, Information Sciences, Vol. 517, pp. 68–85 (2020).
26. Sapp, C. S.: Comparative Analysis of Multiple Musical Performances., ISMIR, pp. 497–500 (2007).
27. Schedl, M.: The LFM-1b Dataset for Music Retrieval and Recommendation, ICMR, pp. 103–110 (2016).
28. Silla Jr, C. N., Koerich, A. L. and Kaestner, C. A.: The Latin Music Database, ISMIR, pp. 451–456 (2008).
29. Simones, L., Schroeder, F. and Rodger, M.: Categorizations of physical gesture in piano teaching: A preliminary enquiry, Psychology of Music, Vol. 43, No. 1, pp. 103–121 (2015).
30. Simones, L. L., Rodger, M. and Schroeder, F.: Communicating musical knowledge through-gesture: Piano teachers' gestural behaviours across different levels of student proficiency, Psychology of Music, Vol. 43, No. 5, pp. 723–735 (2015).
31. Smith, M. and Taffler, R.: Readability and understandability: Different measures of the textual complexity of accounting narrative, Accounting, Auditing & Accountability Journal, Vol. 5, No. 4, pp. 84–98 (1992).
32. Sturm, B. L.: An Analysis of the GTZAN Music Genre Dataset, ACM Workshop MIRUM, MIRUM '12, pp. 7–12 (2012).
33. Whitaker, J. A.: High school band students' and directors' perceptions of verbal and nonverbal teaching behaviors, J. Res. Music. Educ., Vol. 59, No. 3, pp. 290–309 (2011).
34. Zhukov, K.: Teaching styles and student behaviour in instrumental music lessons in Australian conservatoriums, PhD Thesis, University of New South Wales (2005).

Complexity Analysis of Instrumental Performance based on Ontology Structure for Music Selection

Nami Iino[1,2,3] and Hideaki Takeda[1]

[1] National Institute of Informatics
[2] National Institute of Advanced Industrial Science and Technology
[3] RIKEN Center for Advanced Intelligence Project
nami-iino@nii.ac.jp

Abstract. This paper analyzes the complexity of guitar renditions for learners. We have been developing a domain ontology called the Guitar Rendition Ontology (GRO). GRO structurally describes the actual actions of a classical guitar techniques for sharing information and learning the guitar. These descriptions can be used to provide valid information on music for selecting music. Therefore, we first improved GRO because the several renditions lacked of detailed descriptions. Then, we investigated what types of renditions appear in five existing etude books. After that, we analyzed indicators of the complexity of renditions using the GRO's classes and properties. Furthermore, we attempted to calculate the difficulty of each etude by implementing a novel analysis using TF-IDF and our complexity indicators. Experimental results suggested that the difficulty value of an etude corresponds to the creator's subjectivity and intention.

Keywords: Complexity of Action, Music Selection, Guitar Rendition Ontology

1 Introduction and Background

In playing an instrument, the music selected is an important factor. If players can figure out which techniques they are good or bad at and choose pieces appropriately, they can improve their performance. However, it is not easy to take into account a lot of information, such as the actual sounds and body movements, from information on the music alone. Musical pieces and the order that they are in existing instruction books or etudes are empirically selected by composers or players. There are no quantitative indicators at present. We need to start by defining the difficulty as the difficulty appropriate for each player: what is difficult in the first place and what are the factors. It is possible to more accurately analyze difficulty by defining the complexity of the actual actions of specific guitar techniques.

In our previous study, we developed the Guitar Rendition Ontology (GRO). It can serve as a guideline for playing classical guitar on learning and teaching sites [6]. The ontology consists of 96 concepts that describe the relationships between renditions and 18 properties that explain the features of the concepts. We defined three properties for describing the processes of actions for each rendition as the core structure of the guitar rendition concept, that is, action, primary-action, and conditional-action. The descriptions give information on the appropriate way to perform these actions.

In addition, we focused on guitar renditions used in real performances and investigated the trends and patterns of renditions using GRO [5]. However, the number of renditions alone was not sufficient for accurately determining the difficulty of a performance. To overcome this problem, we attempted to evaluate musical compositions quantitatively by defining the complexity of a rendition on the basis of the ontology structure of GRO. This approach enables a new framework that can support music selection for various instruments. Our goal is to provide indicators for selecting music for classical guitar through an analysis of traditional etude books and a discussion.

The organization of the paper is as follows. In Section 2, we mention related works, and in Section 3, we briefly describe the Guitar Rendition Ontology, discuss the problems with it and how to overcome them, and present a new version of it. In Section 4, we first investigate the characteristics of guitar renditions in existing etude books. Then, in Section 5, we present a detailed method for defining the complexities of guitar renditions and analyze the difficulty of etude pieces. In Section 6, we conclude with a summary and overview of future work.

2 Related Works

There are several approaches related to music selection. One of them is music recommendation, and many systems have been designed by using neural network [4], deep learning [3], emotion recognition [9], and so on. Regarding the classical guitar, [7] analyzed guitar pieces from the perspective of information entropy and provided an indicator to support music selection. The situation we are trying to support in this study is that of an instrumental player selecting a piece of music. We thus need to take into account information related to movement that could represent the difficulty of the actual performance.

The field of knowledge processing, several ontologies related to music have been developed: The Music Ontology (MO) for describing metadata about music in detail [12]; Music Theory Ontology for conceptualizing musical and performance symbols in music notation [13]; Two Ontologies focusing on Feedback in music education [16]; Musical Forms and Structures Ontology (MFSO) and Musical Performance Ontology (MPO), which are developed by extending MO, deals with the musical form and its components, as well as the subjective interpretation and advice (emotion, expression, fingering) of the individual [14]. In addition, MPO defines an ”InstrumentTechnique” class that can handle the movements and fingering necessary to realize musical expression. However, it does not specify the style of rendition and the actions involved, which are the important elements of this paper, nor does it discuss the application of them to music selection. Therefore, we believe that our approach is novel and will contribute to the study of instrumental performance.

3 Improving Guitar Rendition Ontology

In this section, we describe the problems with the present Guitar Rendition Ontology (version 2.4) and how to improve it.

Classes and properties: There are important techniques that are not defined or named as common guitar renditions but are used by many advanced players. One of them is *Curve ceja*, a subclass of *Press string rendition*. In this technique, the index finger arches and presses down on the high and low strings, except for the middle string, in order to play only the necessary strings with minimal force. We thus added the *Curve ceja*. Regarding the *Ornament rendition*, we added the following four ornaments that were missing: *Acciaccatura*, *Appogiatura*, *Double appoggiatura*, and *Schleifer*. In the properties, we added "action4" to describe an action in more detail, and we defined "action" as the upper layer of actions 1 to 4.

Until GRO version 2, we classified several renditions by using numbers such as *Cutting1* and *2*, *Tremolo1* and *2*, and *Tune down1* and *2*. However, these numbers cannot characterize each rendition. Therefore, we gave detailed names to these renditions such as *Cutting with right hand* and *by both hand*, *Tremolo by four fingers* and *by one finger*, and *Tune down with right hand* and *with left hand*.

Description of actions: We modified the action descriptions for about 30 renditions. The problem with the previous version of GRO was that the details of the specifications for some of the actions were not enough. For example, for *Tremolo*, the order of plucking with the right finger is usually p, a, m, and i (initials in Spanish, meaning thumb, ring, middle finger, and index finger), so a description up to "action4" is necessary for each finger. However, the previous version described these four fingers by grouping them together in "action1." To overcome this problem, we tried to break down the actions into smaller pieces and describe them (Figure 1). As *Ornament rendition*s and *Figueta* are also techniques that consist of two or more notes, we improved them to describe the action for each note. However, the explanation of which string is actually plucked is not uniquely determined, and this is a subject for future work.

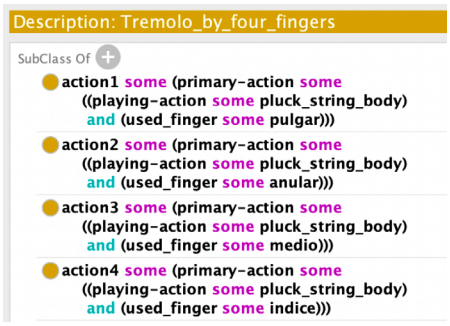


Fig. 1. Description of actions in *Tremolo by four fingers*

Entity / Axiom type	Count
Axiom	1487
Logical axiom	543
Declaration axiom	314
Class	290
-Guitar rendition class	106
Object property	23

Table 1. Ontology metrics of GRO version 3.

Table 1 is a overviews of the improved version[4]. This ontology consists of a total of 313 entities, counting classes, and object properties. The number of classes regarding *Guitar rendition* and the number of properties were increased from version 2.4, from 97 to 106 and from 21 to 23, respectively. Furthermore, axioms that were a combination of

[4] https://github.com/guitar-san/Guitar-Rendition-Ontology

logical, non-logical, and declaration axioms were increased in number by 119 to 1487 by improving the action descriptions in its subclass.

4 Guitar Renditions in Existing Etude Books

This section focuses on learners and investigates what types of renditions appear in existing etude books. There are two cases when making an etude book: (1) books created by the composers themselves and (2) books chosen and organized by performers. Therefore, we tried to identify the differences in renditions when etude books were created from different perspectives.

Since the number of musical pieces varies from book to book, we chose etudes No. 1 to 10 as the subject of our music analysis. We added technique-related information to the musical scores' data and extracted information from MusicXML. Here, one of the authors, who is a guitarist, arranged renditions based on the information already written in the scores.

4.1 Etude books created by composers

The etude books we analyzed are as follows. These were created by two composers from the classical period and a modern composer.

***Estudios Sencillos* by Leo Brouwer:** Leo Brouwer, who was born in Havana in 1939, is a composer, guitarist, conductor, researcher, teacher, and cultural promoter in the modern age. *Estudios Sencillos* [2] is a famous material that has been embraced by many players and by many music schools in their curricula [11].

***25 Etudes Op. 60* by Mauro Carcassi:** Matteo Carcassi (1796-1853) was an Italian guitarist, composer, and pedagogue and is best known for his pedagogical works. His *25 Etudes Op.60* are considered essential works for guitar students. After guitarist Miguel Llobet (1878-1938) added information on fingering, it has been highly regarded as a good teaching tool for learning modern fingering [10].

***12 Etudes Op. 6* by Fernando Sor:** The great Spanish composer and guitarist Fernando Sor (1778-1839) is known for his many guitar compositions. His opus numbers range up to about 63, and these consist of many solo pieces, guitar duets, and songs and guitar pieces. We chose his opus 6, which is known as advanced grade, from several etudes to compare the above books.

Figures 2 to 4 indicate the number of renditions to each etude and the information entropy calculated from them. The etudes of Brouwer and Carcassi have a similar tendency of using *Full planting* in early numbers and *Descending slur* and *Ascending slur* in late numbers. In comparison, the etudes of Sor are structured in such a way that these are used alternately.

Carcassi's etude is often used most on a lesson sites among the above three etudes. The edition by Yasumasa and Seiko Obara, which is often used in Japan, notes that "There are 25 pieces in this collection, and we would like you to follow them in order. The reason is that this is one composition consisting of 25 pieces, all of which are related to each other in key. When you can play all 25 pieces correctly from memory,

you are considered to have achieved a certain level of perfection, hence it is not desirable to practice only one or two pieces" [10]. This suggests that the composer arranged the pieces with the same intention, as the order of the pieces is an important factor in learning the guitar. In fact, Figure 3 demonstrates an increase in both the number of renditions and the information entropy.

In *Estudios Sencillos*, Brouwer stated that "This is the beginning of a series of etudes that were composed for the real guitar apprentice. Every technical problem is separated by the degree of difficulty of the rest of information. If, for instance, there is an arpeggio for the left hand, we are going to do it so that the other hand, in this case the right one, does not find much problem." This suggests that he was as meticulous in composing and structuring his etudes as Carcassi, or even more so. At least, his etudes showed a similar trend to Carcassi's than to Sor's.

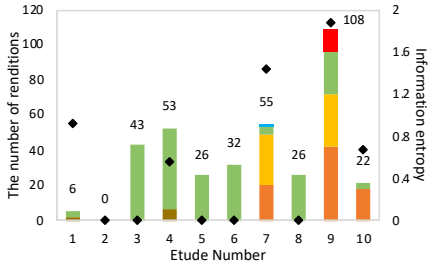


Fig. 2. Change of renditions in *Estudios Sencillos* by Leo Brouwer.

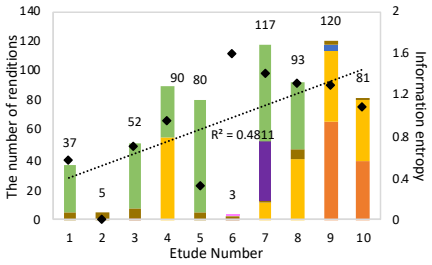


Fig. 3. Change of renditions in etude op. 60 by Matteo Carcassi.

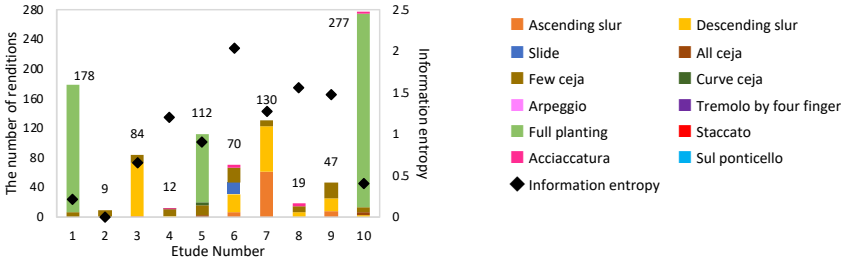


Fig. 4. Change of renditions in etude op. 6 by Fernando Sor.

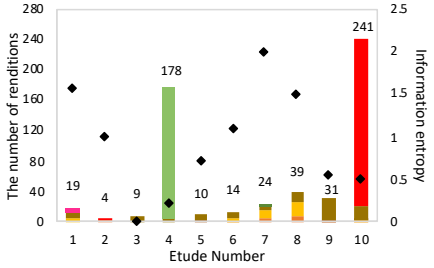


Fig. 5. Change of renditions in Sor's etudes by Andrés Segovia.

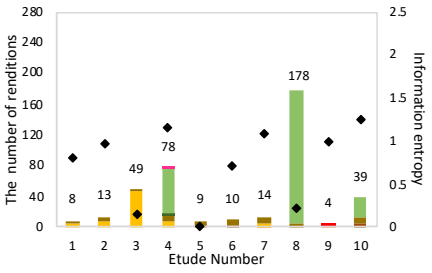


Fig. 6. Change of renditions in Sor's etudes by Yasuo Abe.

4.2 Etude books created by players

We took up Fernando Sor, the composer described in the previous section, and analyzed two etude books collecting his works. These books were published by famous guitarists, and they include pieces other than Op. 6, regardless of the opus number. The data was extracted on the basis of the fingering of each guitarist written in the score.

Twenty Studies by Andrés Segovia: Master musician Andrés Segovia (1893-1987) has selected 20 of Sor's etudes and edited them in a rational order with instructions on fingering, conception, and speed [15].

***25 Etudes* by Yasuo Abe:** This book, which was published by Yasuo Abe (1925-1999), is a collection of Sor's most important etudes for learning all the advanced techniques of the guitar [1].

As shown in Figures 5 and 6, the result differs from Sor's *12 Etudes Op. 6* in that the overall number of renditions is small but that of a few etudes is extremely large. According to Segovia, *Twenty Studies* can be used not only for improving students' technique, but also for maintaining advanced players' technique to a certain level [15]. It includes arpeggio, chords, legato, left hand fingerings, ceja exercises, and many more forms.

Concerning the order of the pieces, Abe noted that "This book is arranged in an easy-to-follow order, which is not the same as Sor's opus numbers," and "Please practice in order from the beginning." This means that the order of the pieces is an important factor in this book. However, it was difficult to identify a specific pattern from these graphs that indicates the above intents. This suggests that, unlike the composer, the players made their selections and ordered the pieces on the basis of their very subjective impressions when playing the pieces.

5 Analyzing Complexity of the Performance

Each guitar rendition requires a different set of actions. The load on the player is different when using a rendition that requires multiple actions or when using a rendition with simple actions. In other words, the complexity of an action relates to the difficulty of the performance. Therefore, we clarified the complexity of each rendition on the basis of the description of actions in the GRO in this section. Here, we analyzed renditions that are located at the bottom of the *Guitar Rendition* Class hierarchy and contain descriptions of actions.

5.1 Complexity of guitar rendition

The calculation rules for weighting properties and classes are as follows.

1. Action Related Properties: Weight of 1 to each of the following properties that represent the relationship of the actions: "action1," "action2," "action3," "action4," "conditional-action1," and "conditional-action2." We excluded "primary-action" and "playing-action" because actions 1–4 contain them.

2. **Detailed Properties:** Some actions have restrictions or requirements in describing the actions in detail such as "direction," "number," "place of action," "part of hand or finger," "timing," "used finger," "used string," "tool," and "ornament tone." We believe that these are related to the complexity of an action. Therefore, the number of properties described in each rendition was additionally weighted.
3. **Player's Action:** It is not only the relationship between actions, but also the type of movement that is important. In the GRO, 22 actions are defined in the class called *Guitar player's action*. We weighted these actions on a scale of 0 to 3. For example, "pluck string" and "touch string" are 0, "press string" and "pick up string" are 1, and "rub string" and "cross strings" are 2. Regarding "use slur," we used the highest value of 3 because *Slur* is defined as a rendition. These values are arbitrary and can be individually adapted to the player.

Figure 7 shows a treemap chart with the complexity value of each rendition represented by the area of the rectangle. The range of values is from a complexity of 1 to 16. *Turn with left hand*, a subclass of the *Ornament rendition*, showed the largest value 16, and other renditions in the same category also tended to map high values. In comparison, the *Fingering rendition* values tended to be low, ranging from the lowest value of 1 for *Al aire* to 8 for *Figueta*, because this is a basic technique of classical guitar. *Percussive rendition* was also similar, with values ranging from 3 to 8. Moreover, all renditions of *Note value rendition* were low at 2.

The values of *Articulation rendition*, *Chord rendition*, *Pitch change rendition*, and *Timbre rendition* varied. The reason for the large difference in values between both *Slur* (also *Slide*) and *Tapping* for the *Articulation rendition* is that the former is a technique learned at the beginning stage, while the latter is an applied technique and uses the attribute "use slur" in GRO. For *Chord rendition*, which requires techniques using multiple strings, *Tremolo* and *Rasgueado* had a large area. *Tremolo* is an especially difficult technique that even some professional guitarists are not good at.

The 12 renditions, extracted in the previous section are represented by black boxes. As mentioned above, *Tremolo* had the largest value because it is a difficult technique. *Ceja* related renditions were large because they required advanced techniques that are performed by pressing multiple strings with a single finger. It is reasonable that the *Descending slur*, which is played by hooking and plucking the string with the left finger, was larger than the *Ascending slur*, which is played just by tapping the string with the left finger. *Descending slur* and *Acciaccatura* are the same action, so the complexity values are equivalent. *Staccato* and *Sul ponticello* were small because they require the simple action of muffling or changing the plucking position with the right finger.

From these results, we found that the complexity indicator corresponds generally to the intuitive difficulty of a rendition. Applying these complexity values to musical pieces can be provided an effective indicator for music selection. However, there is a problem with a few of the rendition values. *Full planting* is a simple technique where the fingers are set before plucking, so the value should be lower than that of *Arpeggio*. We will discuss this further towards a more complete complexity calculation that matches the player's intuition.

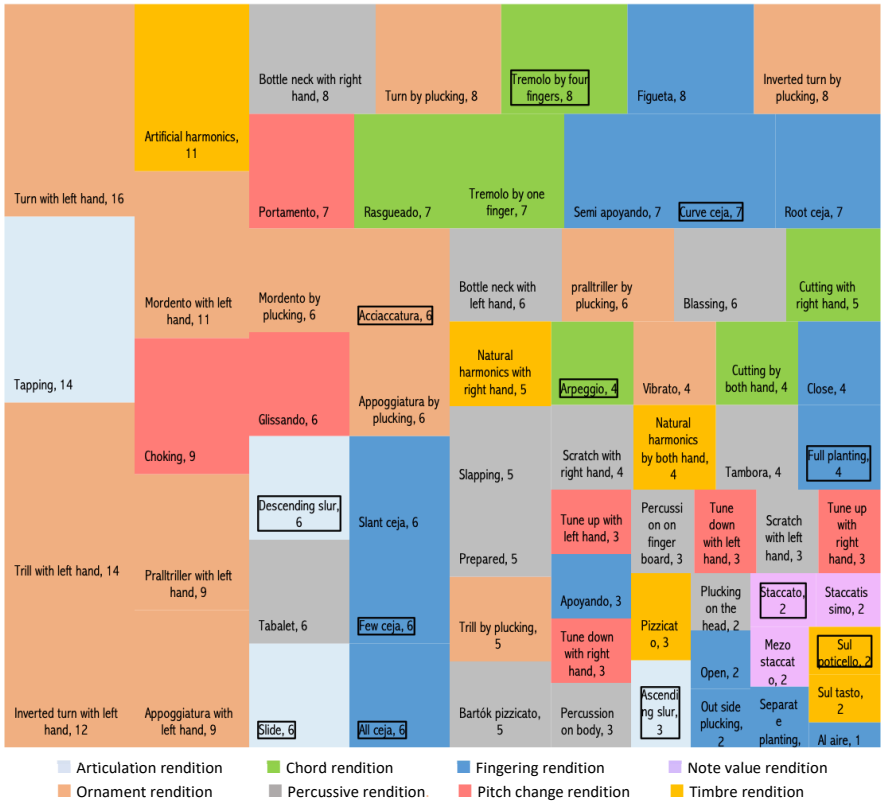


Fig. 7. Treemap of complexity of each guitar rendition.

5.2 Analyzing difficulties of etudes

In regarding to guitar renditions, we think that there are three types of difficulty when playing a musical instrument:

1. Difficulty of the rendition itself
2. Difficulty with the number of renditions
3. Difficulty with the order of the renditions

In this study, we considered the complexity value of a rendition (defined as $complexity_{r,e}$) presented in Section 5.1 as an indicator of type 1. In addition, we attempted to extract indices related to types 2 and 3 by calculating TF-IDF (Term Frequency Inverse Document Frequency) [17] by focusing on the number of occurrences of a rendition. The TF-IDF value is expressed by the following formulas:

$$\text{TF-IDF} = tf_{r,e} \cdot idf_r \qquad tf_{r,e} = \log \frac{n_{r,e}}{\sum_k n_{k,e}} + 1 \qquad idf_r = \log \frac{|D|}{|\{d : t_r \in d\}|} + 1,$$

where $n_{r,e}$ is the value obtained by weighting $complexity_{r,e}$ to the rendition frequency for an etude f_e, $\sum_k n_{k,e}$ is the total number of renditions (including the weight of $complexity_{r,e}$) in the etude, $|D|$ is the total number of etudes in a corpus, and $\{d : t_i \in d\}$ is the number of etudes that contain at least one rendition. Furthermore, the difficulty level of each etude, which is expressed by $difficulty(e)$, is calculated from the following formula:

$$difficulty(e) = \sum_{k \in e} \text{TF-IDF}_{k,e}.$$

Figure 8 indicates the difficulty level of each etude number for the five etude books analyzed in Section 4. Although Carcassi's etude was relatively high and Brouwer's etude was low, both of them had a tendency for the difficulty level to increase: R^2=0.470 and R^2=0.276. Moreover, Abe's etude showed the similar tendency of the line as Carcassi's etude (R^2=0.261). These etudes are popular pedagogical materials around the world. As described in Section 4, it is clear that they took into account the order in which learners can easily practice. For the etude books made by Sor and Segovia, we could not determine whether the difficulty level corresponded to the etude number because the graphs remained almost unchanged. In fact, there are no explicit instructions in those two books regarding the order of the pieces. Therefore, we found that our approach was somewhat consistent with the subjectivity and intentions of the creator (i.e., the author of the etude book). We need to increase the number of analyzed musical pieces and conduct a subjective evaluation of the players using the results obtained in this study.

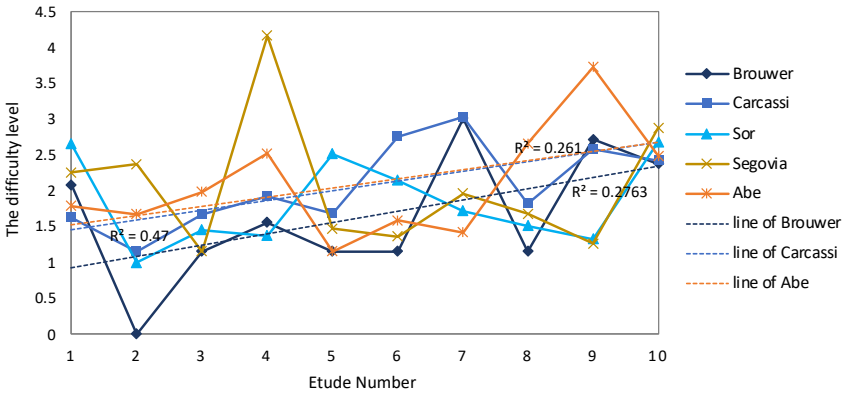


Fig. 8. Difficulties of etudes in five books.

6 Conclusion

In this study, we presented an approach to providing an indicator of the complexity of a guitar rendition and the difficulty of a piece on the basis of the Guitar Rendition Ontology (GRO). We first modified the GRO to describe detailed actions of renditions. Second, we investigated the number of renditions in existing traditional etude books.

Third, we analyzed the renditions' complexities and visualized them with a treemap. Finally, we calculated the difficulty for each etude by using TF-IDF and complexity indicators. As a result, we found that the etude number in etude books corresponded to the subjective perceived difficulty of the creator. The contribution of this paper is to propose a novel approach that quantitatively measures the difficulty of music itself by using a complexity indicator calculated on the basis of the ontology structure of GRO. The advantage of our method is that it is individually adaptable. As a future work, we will construct a framework that will enable music to be selected from more diverse perspectives.

References

1. Abe, Y.: Fernando Sor 25 Etudes, ZEN-ON Music co., Ltd. (1966)
2. Brouwer, L.: Guitar Works for guitar solo, Max Eschig (2006)
3. Fessahaye, F. et al.: T-RECSYS: A Novel Music Recommendation System Using Deep Learning, 2019 IEEE International Conference on Consumer Electronics (ICCE), pp.1–6, (2019)
4. Hansen, C. et al.: Contextual and Sequential User Embeddings for Large-Scale Music Recommendation, In Fourteenth ACM Conference on Recommender Systems (RecSys'20). Association for Computing Machinery, pp.3–62 (2020).
5. Iino, N., Hamanaka, M., Nishimura, T,. K., Takeda H.: Music Analysis with focusing on Techniques by using the Guitar Rendition Ontology, Vol.2019-MUS-124 No.13, (2019)
6. Iino, N., Nishimura, S., Nishimura, T., Fukuda, K., Takeda H.: The Guitar Rendition Ontology for Teaching and Learning Support, The 13th IEEE International Conference on Semantic Computing (ICSC), Resource track, Vol. 1, pp.404–411 (2019)
7. Iino, N., Iizuka, Yasuki., Okino, Shigeki.: Study on Performance Program in Classical Guitar Competitions for Supporting Piece Selection, IPSJ Journal, 59(3), pp.904–911, (2018)
8. Lim, K. A. and Raphael, C.: InTune: a musician's intonation visualization system, ACM SIGGRAPH (2009)
9. Maningo, J.M.Z. et al.: A Smart Space with Music Selection Feature Based on Face and Speech Emotion and Expression Recognition, 2020 IEEE REGION 10 CONFERENCE (TENCON), pp.696–701 (2020)
10. Obara, Y., Obara, S.: 25 Estudios Op.60 Matteo Carcassi, Edition Casa de la Guitarra, No.101 (1965)
11. Penaranda, C., Isaac, C.: Leo Brouwer's Estudios Sencillos for Guitar: Afro-Cuban Elements and Pedagogical Devices, Dissertations, 1002 (2009)
12. Raimond, Y., Abdallah, S.A., and Sandler, M.B., Giasson, F.: The Music Ontology, Proceedings of the 8th International Conference on Music Information Retrieval (2007)
13. Rashid, S.M., Roure, D.D., McGuinness, D.L.: A Music Theory Ontology, Proceedings of the 1st International Workshop on Semantic Applications for Audio and Music, pp.6–14 (2018)
14. Sébastien, V., Sébastien D., Conruyt, N.: Annotating works for music education: propositions for a musical forms and structures ontology and a musical performance ontology, Proceedings of the International Conference on Music Information Retrieval (2013)
15. Segovia, A.: Twenty Studies for the guitar by Fernando Sor, EMI Music Publishing Japan Ltd. (2000)
16. Yee-King, M.J., and Wilmering, T., Rodriguez, M.T.L., and Krivenski, M., d ' Inverno, M.: Technology Enhanced Learning: The Role of Ontologies for Feedback in Music Performance, Frontiers in Digital Humanities, Vol.5, No.29 (2019)
17. Zhang, W., Yoshida, T., Tang, X.: A comparative study of TF*IDF, LSI and multi-words for text classification, Expert Systems with Applications, Vol.38, No.3, pp.2758–2765 (2011).

Music

Approaches
Karl F. Gerber
for flute and interactive violin automaton
Karina Erhard, flute

Miskets and Canicas
Anıl Çamcı and Matias Vilaplana
audiovisual networked piece for two performers playing a Virtual Interface for Musical Expression (VIME)

Xeno
Enrico Dorigatti
fixed multimedia

Time Garden: dawn replica
Charles Nichols, Zach Duer and Scotty Hardwig
computer music accompanying video of motion capture dance in virtual reality

Water
Christian M. Fischer
fixed multimedia

TeleFAUXcus
Sarah Hamilton, Bradley Robin and Seth Shafer
networked performance

Construction in Kneading
Ryo Ikeshiro
live generative audiovisual work

anthozoa
Daniel Blinkhorn
multichannel fixed media

Mojave
Paulo C. Chagas and Cássia Carrascoza
audiovisual composition for flute, electronics and 3D-video
Cássia Carrascoza, flute

Things I Have Seen in My Dreams
João Pedro Oliveira
fixed multimedia

Cat-Dog
Greg Beller
musical theater
Lin Chen and Lini Gong, performance
Janina Luckow aka KLARA, video

It was a calm and relaxing night
Francesco Corrias
for violin and live electronics
Francesco Fadda, violin

Mechanophore
Scott Barton
for virtual and robotic strings and percussion

YAMAHA
Make Waves

GIVING UP WHAT I LOVE MOST IS NOT AN OPTION FOR ME MAKE WAVES.

Pianist Hiromi Uehara

Hiromi

#YAMAHAMAKEWAVES

YAMAHA CORPORATION