Craig T. Jin, Matthew E.P. Davies, and Patrizio Campisi

Embedded Systems Feel the Beat in New Orleans

Highlights from the IEEE Signal Processing Cup 2017 Student Competition

Foot-tapping and moving to music is such a natural human activity, one may assume that feeling the beat in music is a simple task. Feeling the beat and then producing it, e.g., by foot tapping, is an intrinsically real-time process. As listeners, we do not wait for the beat to occur before tapping our foot; instead, we make predictions about when the next beat in the music will occur and continually revise our sense of the beat based on the accuracy of our predictions. Likewise, performing musicians have shared sense of beat, which is what allows them to play in time together.

This type of high-level music listening and understanding sits at the heart of the challenge set for this year's IEEE Signal Processing Cup (SP Cup) competition, the final stage of which concluded at the 2017 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), hosted in New Orleans, America's jazz heartland, on 5 March. The participating undergraduate cohort had to devise and construct a creative, embedded application demonstrating a real-time response to the beat of the music. Depending on the genre, composition, and rhythmic complexity of a musical piece, real-time beat tracking poses considerable challenges, which are equally present for human listeners, especially those without formal musical training. Throughout the SP Cup, the teams confronted these challenges from both the human and computational perspectives via the choice of training and testing material, the human annotation of beat locations, the implementation and evaluation of their beat-tracking algorithms, and the response to the beat in their creative applications.

Digital Object Identifier 10.1109/MSP.2017.2698075
Date of publication: 11 July 2017

Beat tracking in music signals

The task of beat tracking of music signals has been an active area of music signal processing research for more than 25 years. While many of the earliest computational approaches sought to emulate the human process of tapping the beat in real time by making predictions of future beats [1], [2], a marked shift occurred in the early-to-mid 2000s toward offline approaches that could observe the entire musical input prior to determining beat locations.

The standard pipeline for offline beat tracking involves the explicit identification of note onset locations (or an "onset strength function," which emphasizes their location) that are subsequently passed to a tempo-estimation stage used to estimate the latent beat periodicity in the input signal, followed by the recovery of the phase (or alignment) of the beats to the music. Common techniques used to extract the beat from music signals include multiagent systems, dynamic programming, hidden Markov models, and a mixture of experts systems. Current state-of-the-art methods employ deep neural network architectures to learn the relationship between labeled beat annotations in training data sets and feature representations extracted from musical audio signals, thus leveraging both advanced signal processing and machine learning.

The growth of offline approaches arose in part by the significant increase in the use of beat tracking for so-called beat-synchronous analysis as an intermediate processing step within other music signal analysis tasks, such as structural segmentation, chord detection, and music transcription. With the shift toward making multiple passes across input signals, the focus on real-time analysis was reduced. Furthermore, with a greater emphasis on the accuracy of beat tracking over computational efficiency, offline approaches also provided the opportunity for tracking the beat in music with expressive timing (i.e., changes in tempo) something that was considered impossible for real-time systems bound by the need to make predictions of future beats in the music [3].

An emerging topic related to the domain of music signal processing is creative music information retrieval, which seeks to open new possibilities for music creation, interaction and manipulation. This is facilitated by the robust analysis and interpretation of music signals [4]. For applications that target live interaction between users and/or musicians and technology, there is a compelling need to perform music signal analysis in real time. One specific motivation for the SP

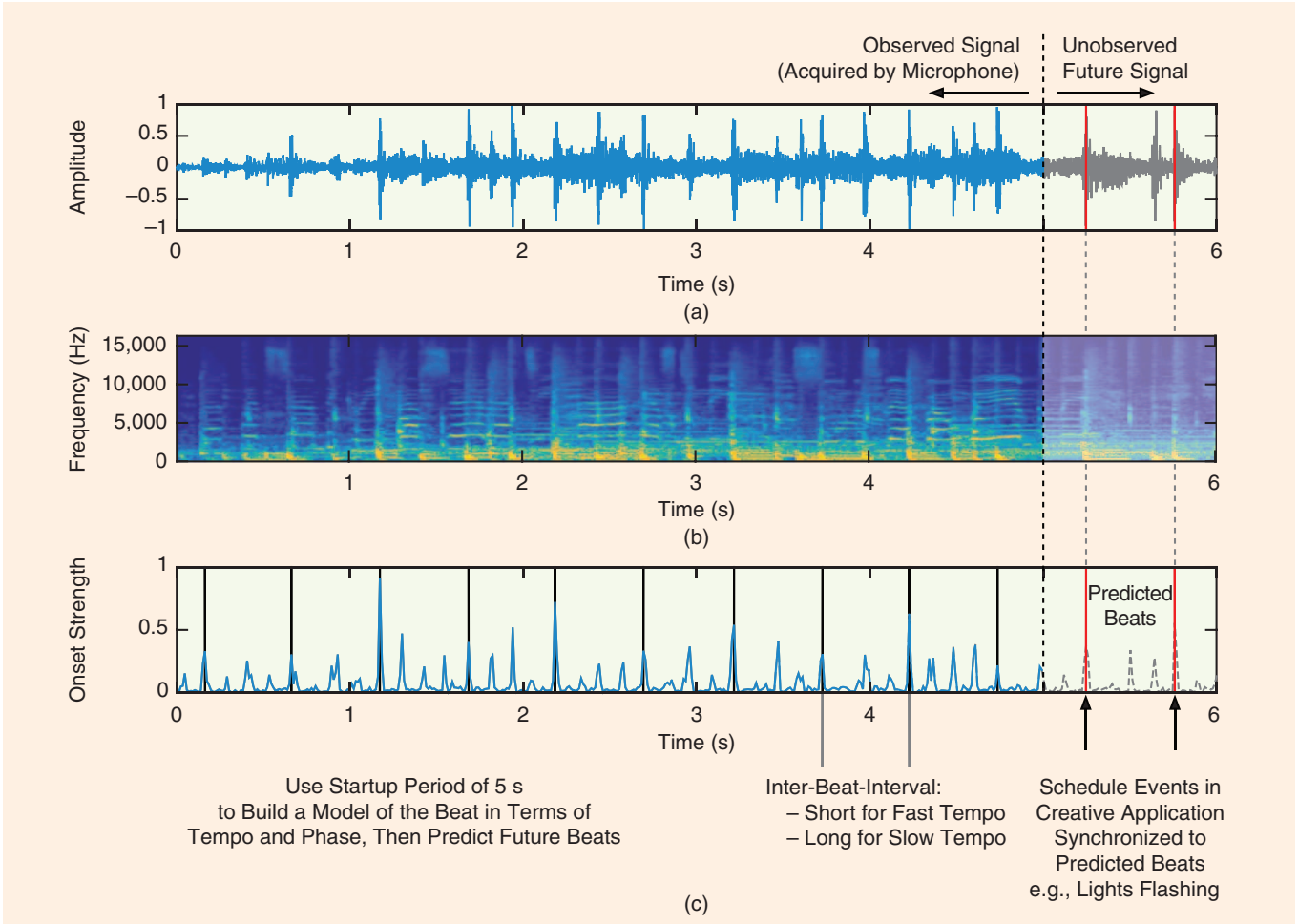


FIGURE 1. An overview of real-time beat tracking. (a) An input audio signal for which the first 5 s have been acquired by a microphone. (b) A spectrogram representation of the input audio signal used to generate the onset strength function. (c) The onset strength function with overlaid beat estimates shown in black and predicted future beats in red.

Cup was, therefore, to reimagine research into beat tracking with an explicit link to real-time creative applications. From a technical perspective, real-time beat tracking, unlike offline approaches, must extract an onset strength function, estimate tempo and predict future beats based only on a continuously evolving observation of the input signal, and thus it sits firmly at the more challenging end of the spectrum. This real-time requirement also imposes strict computational limitations, a difficulty that is only increased by constraining the use of hardware in the SP Cup to embedded devices with limited computational resources. The final aspect of the competition—developing a creative application that reacts to the (predicted) beat of the music—provides an open-ended activity for the teams, but one that must be also performed in real time on the embedded device. An overview of the process of real-time beat tracking is shown in Figure 1.

The SP Cup is an undergraduate competition organized by the IEEE Signal Processing Society (SPS) in which undergraduate students work in teams to tackle a real-life signal processing problem. Launched in 2014, the SP Cup competition has been held annually, and 2017 is the fourth edition.

To join, undergraduate students are required to form a team. Each team is composed of one faculty member to advise the team members, up to one graduate student to assist the supervisor in mentoring the team, and three to ten undergraduate students. Three top teams are selected from the initial round of competition and provided travel grants to participate in the final competition at 2017 ICASSP. The final results are described in "Winners of the SP Cup 2017."

Tasks in the SP Cup 2017

The SP Cup challenge covered the many and multidisciplinary aspects of beat tracking, with the aim of giving students training in several areas such as music understanding and beat annotation, strategies for selecting content for training and competition, signal processing, computational optimizations for real-time performance, hardware implementations, and creative application design and development. With such a wide range of tasks and challenges to address, the SP Cup 2017 was seen as the most challenging edition so far. All of the resources related to the competition can be found at http://sydney.edu.au/engineering/electrical/carlab/beatracking.htm.

The open competition stage

The SP Cup started with an open competition stage from June 2016 to January 2017, consisting of two parts. The

objective for part one was to submit three 30-s musical excerpts with human-annotated beat times. The judging criteria was the quality of the beat annotations. For the first part, participants were provided with a database of 50 musical excerpts spanning a range of styles and difficulties. The database was split into two halves. One half was open, meaning that for these musical excerpts, human-annotated beat times were provided. The other half was closed so that the annotated beat times for these musical excerpts remained hidden. The purpose of the database was to assist with the development and testing of real-time beat-tracking algorithms. The task for the first part consisted of an exercise in crowdsourced beat annotation. Each participating team was required to provide human-annotated beat times for three musical excerpts of their own choosing. They also nominated one of the three musical excerpts as a challenge piece so that the beat annotations would remain hidden from the other participating teams.

To assist participants with the evaluation of their beat-tracking algorithms and give a reference for how beat-tracking accuracy would be calculated, a MATLAB evaluation script was provided. The evaluation method, extended from [5], gives an accuracy score based on a comparison of estimated beat times with annotated ground truth. It calculates the proportion of continuously correct beat estimates occurring with a perceptually specified tolerance window around the ground truth annotations. To mirror the ambiguity in human perception of the beat in music, estimated beats at perceptually related metrical levels to the ground truth annotations (e.g., twice or half the tempo of the ground truth for music in 4/4 time) were also considered correct.

The first part of the open competition was devised to serve multiple purposes. From the perspective of the teams wishing to participate, the annotation of three musical excerpts provided a relatively low barrier for entry, while also offering teams the chance to actively shape the SP Cup through their personal choice of musical content. For the organizers, the use of team submitted content led to the creation of a totally new annotated data set for beat tracking (free from sampling bias) and, furthermore, one that could reflect the cultural diversity of the teams who participated.

For the second part of the open competition, participants had to develop and implement their beat-tracking algorithm on an embedded device (the choice was left open, but most used the Raspberry Pi for beat tracking and an Arduino for control of the output) so that it achieved real-time performance. The objectives for part two were the following:

1) real-time embedded software with instructions on how to run it
2) beat-time output for the real-time embedded device for the database and participant submitted musical excerpts
3) a video demonstrating real-time operation
4) a report in the form of an IEEE conference paper.

The judging criteria were a performance score for the real-time embedded algorithm and a creative application score. Participants then had to design and construct a creative application for their real-time beat-tracking device. In addition to submitting the beat-tracking output of their systems across all of the available musical material as well as providing source code with installation instructions, participants also had to submit a report in the form of an IEEE conference paper and post a video online demonstrating the creative application and real-time operation. This year's SP Cup is unique in that the competition included real-time constraints as well as a creative application.

The teams were evaluated on three main components submitted across both parts of the open competition. In the first part, a team of experts active in beat-tracking research assessed the subjective quality of the annotations and made corrections where necessary so as to ensure their validity as ground truth. In the second part, the submitted beat times provided by each team on the musical material without released annotations were evaluated using the publicly available MATLAB script. In addition, the creativity of the demonstrated applications were assessed, again by a group of experts. Since each team submitted the beat-tracking software for their real-time embedded device as part of the submission for the open competition, the real-time operation and its beat-tracking output could be verified. The final score for each team was weighted across these three components with the following proportions: one-sixth for the annotations, one-half for the real-time beat-tracking accuracy, and one-third for the creative application. A breakdown of the scores as well as a written assessment by the organizing committee was provided to all teams that participated in the second part of the open competition.

Final competition

After the judging committee evaluated the submissions from the open competition, three finalist teams were chosen to advance to the final competition. Prior to attending the final event at ICASSP, each team was required to submit additional annotated challenge excerpts to be used for on-site evaluation. However, in contrast to earlier stages in this year's competition, neither the audio nor the annotations were made available to the other teams.

The final SP Cup event was held at ICASSP in New Orleans, Louisiana, on 5 March. For the first time since the inception of the SP Cup, a live demo session was included in the final event. The event started by testing the accuracy of the real-time beat-tracking embedded devices in real-world conditions with the audio of the newly submitted challenge pieces captured by microphones (Figure 2). The finalist teams were then allowed time to set up their live demos. Each team then presented its beat-tracking algorithm, its implementation, and the design and development of the creative application. This was followed by a live demonstration of the creative application and a question and answer session. The final judging committee convened and selected the first-, second-, and third-prize winners as well as presented honorable mentions.

Grand Prize: Team Beats on the Barbie

- University of New South Wales
- Undergraduate students: Angus Keatinge, Max Fisher, Jeremy Bell, and James Wagner
- Supervisor: Vidhyasaharan Sethu
- Video: https://www.youtube.com/watch?v=VkoGZnVEsfw
- Technical Approach: Team Beats on the Barbie (Figure S1) adapted and optimized an existing real-time beat-tracking algorithm [6] for Rasberry Pi. They controlled their creative application, a robotic drumming system (see Figures S2 and S3), using an Arduino Mega. The robotic drummer can play back a drum part encoded as an Arduino sketch, and during the final competition it accompanied team members Jeremy Bell and James Wagner in a performance of John Lennon's "Imagine." Due to the use of high-powered solenoid drivers and fast triggers, the system was able to play drum fills and was loud enough to require no additional amplification.

Second Prize: Team Madmom

- Johannes Kepler University, Austria, and Télécom ParisTech, France
- Undergraduate students: Amaury Durand (Télécom ParisTech), Sebastian Pöll (Johannes Kepler University), and Raminta Balsyte (Johannes Kepler University)
- Supervisor: Sebastian Böck
- Graduate Mentor: Florian Krebs
- Video: https://www.youtube.com/watch?v=Losv4GqsGYU
- Technical Approach: Team Madmom (Figure S4) adapted a real-time beat-tracking system from the existing offline approach in the Madmom Python library [7] and used a recurrent neural network. To allow real-time operation, the bidirectional neural network was replaced with a unidirectional network. They controlled their creative application, a robotic drumming system (see Figures S5 and S3), using a Raspberry Pi. Instead of a preprogrammed drum pattern, the system inferred what to play based on the analysis of the rhythmic

FIGURE S1. First place team: Beats on the Barbie.

FIGURE S3. The automated drums for Team Madmom and Team Beats on the Barbie. Both teams implement drum signals for the bass drum, the snare drum, and the hi-hat.

FIGURE S2. Solenoid-based actuators for Team Beats on the Barbie.

FIGURE S4. Second place team: Team Madmom.

structure of the input and was able to react to changes in a time signature. Team Madmom intends to make its system freely available and open source at https://gitlab .cp.jku. at/ROBOD.

Third Prize: Team PulseBox

- University of Maryland, United States
- Undergraduate students: William Heimsoth, Creed Gallagher, and Josh Preuss
- Supervisor: William Hawkins
- Video: https://www.youtube.com/watch?v= KPwFnY6bJNI
- Technical Approach: Team Pulsebox (Figure S6) developed all aspects of their system entirely from scratch. Their beat-tracking algorithm made use of a novel comb-snapping technique to maintain high temporal accuracy of the predicted beats and used machine learning to optimize multiple relevant parameters including those related to tempo adjustment, windows, and the choice of frequency bands. Their creative system, the PulseBox, (shown in Figure S7) was a light-emitting diode (LED) cube containing 245 LEDs arranged in a 7x7 grid on each of the five visible faces of the cube. The LEDs were individually configurable with 24-bit color and were programmed to react to the beat of the music with rotating shapes and patterns.

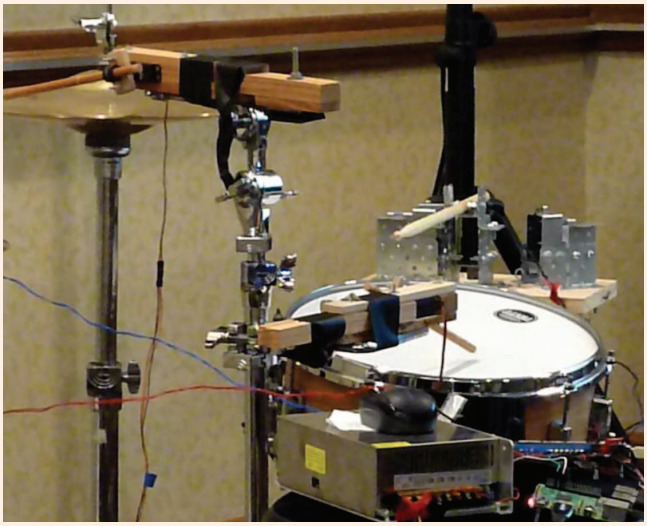


FIGURE S5. The drum actuators for Team Madmom.

In addition to the three overall winning teams (Figure S8), the SP Cup 2017 judging committee made the following honorable mentions. Videos for these and other submissions can be found at http://sydney.edu.au/engineering/electrical/carlab/beatracking.htm.

Honorable Mention for Excellent Video Production and an Entertaining Concept

- Team NTHU-EECS, National Tsing Hua University, Taiwan.

Honorable Mention for Excellent Video Production and Accurate Ground Truth Annotation

- Team Impulse, Bangladesh University of Engineering and Technology, Bangladesh.

Honorable Mention for Excellence in Ground Truth Annotation and Beat-Tracking Performance

- Team Sharif University of Technology, Sharif University of Technology, Iran.

FIGURE S6. The third place team: Team PulseBox.

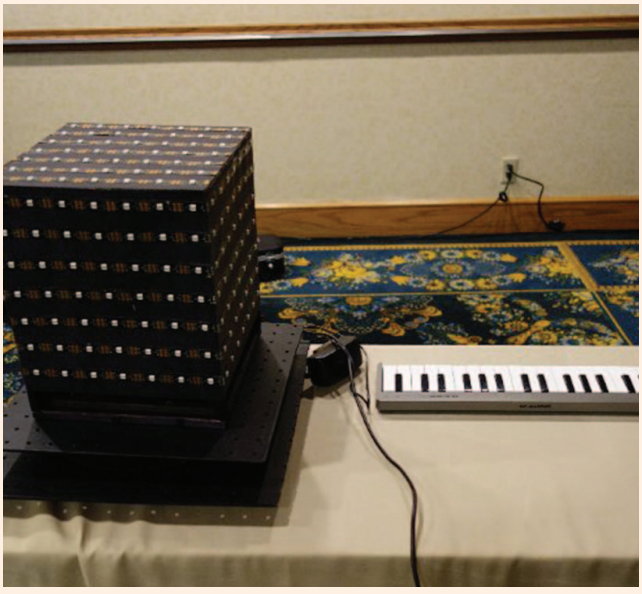


FIGURE S7. The rhythmic LED cube for Team Pulsebox.

(a) (b) (c) (d) (e)

FIGURE S8. A behind-the-scenes look at the SP Cup 2017 teams that received honorable mentions: (a) Team Sharif, University of Technology, Iran; (b) and (c) Team NTHU-EECS, Taiwan, with their metronome mechanism; (d) and (e) Team Impulse, Bangladesh, and their band of skeletons.

Highlights of technical approaches

For the real-time beat-tracking aspect of the SP Cup, many teams implemented methods inspired by or directly adapted existing approaches for beat tracking. Even in the cases where a reference implementation was publicly available, these required a very significant overhaul to make the algorithms real-time compatible and sufficiently optimized to run on the embedded devices.

From an algorithmic perspective, the great majority of submitted algorithms followed the standard approach for beat tracking by

- generating one or more onset strength signals (often across subbands) derived from time-frequency representations of the streaming input audio signal
- performing periodicity analysis on the onset strength signal(s) by means of autocorrelation or comb filtering
- estimating the phase of the beats by cross-correlation or dynamic programming, and then using phase as the reference point from which to predict future beat locations.

Many teams also included some higher-level modeling to provide a smooth output without rapid switching between metrical levels (i.e., tempo doubling or halving). Depending on the computational resources of the chosen embedded device (some of which were extremely low power), the beat-tracking approach had to be highly optimized, e.g., purely based on time-domain analysis. The most computationally expensive and ambitious approaches attempted to run state- of-the-art deep neural network architectures for beat prediction.

The technical approaches were invariably biased by the initial project description, which mentioned blinking LEDs and the Raspberry Pi and Arduino. So, for example, the Raspberry Pi was the embedded platform used for beat tracking by the majority of teams (fourteen teams). A variety of other interesting embedded platforms were used by a single team: ARM mbed, NAO robot, STM32F4Discovery, and UDOO Quad. Many teams coupled the beat tracker with an Arduino to assist with the creative output. With regard to the programming language used for the embedded application, it was evenly distributed between C/C++ and Python. A wide variety of creative applications were demonstrated. Applications demonstrated by multiple teams were: LED displays (seven teams); screen displays (four teams); and automated drumming (two teams). The unique creative applications were a moving head, a dancing robot, a band of skeletons, a metronome follower, a vibration device for the hearing-impaired, and an encryption device.

SP Cup 2017 Statistics

In total, the teams from 20 different countries participated in SP Cup 2017. At the registration stage of the competition, 40 teams were involved with a total of 279 participants. For the first part of the open competition, 33 teams across 18 countries with more than 250 participants submitted musical excerpts (thus adding 99 new examples to the initial data set of 50 provided by the organizers). In the second part, which presented a significant increase in difficulty and submission requirements, 21 teams participated with 147 members spread across 14 countries. The countries with the most registrations were India with eight and the United States with seven.

As in previous years, the SP Cup was run as an online class on the Piazza platform, which, in addition to allowing continuous interaction with teams, also hosted the test material supporting documentation. In total, 115 students registered for the course, with approximately 220 contributions and 2,500 views of the posts. An archive of the class is available at https://piazza.com/ieee_sps/other/sp1701/home.

Since its inception, the SP Cup has received generous support from MathWorks, Inc., the maker of the popular MATLAB and Simulink platforms. MathWorks also provided funding support to the SP Cup and contributed their expertise. Each student team that registered for the SP Cup was provided complimentary software access to MATLAB and related toolboxes. After discussion with the SP Cup organizers, MathWorks provided skeleton code for real-time audio using a Raspberry Pi, which is available at http://au.mathworks.com/matlabcentral/fileexchange/59825-real-time-beat-tracking-templates-for-ieee-signal-processing-cup-2017. The IEEE SPS welcomes continued engagement and support from industry in future SP Cup competitions. Interested supporters may contact Dr. Patrizio Campisi, director for student services, at patrizio.campisi@uniroma3.it.

Participants' feedback

Throughout the open competition there was a great deal of interaction, not only through questions for the instructors posted to Piazza but also among the different student teams who often engaged in discussion over the provided responses. Indeed, these interactions were critical in expanding the flexibility of the evaluation script to correctly process music in non-4/4 meters. As organizers, we were delighted to see this collaborative spirit continue right through to the preparation for ICASSP and the final session itself. Next, we provide an overview of some feedback and perspectives received from the three winning teams.

FIGURE 2. A Real-time beat-tracking assessment for the final competition: music was played from the Bluetooth loudspeaker and recorded by three microphones, one for each team.

Team Beats on the Barbie

- "The project itself was extremely challenging. I worked on the software implementation of the algorithm, and to do this meant implementing the hardware interface on an embedded system. For me, the most challenging part of the SP Cup was setting up many different projects and libraries that often had never been tested on an embedded system to work in real time and simultaneously. This required running parts of the algorithm in different threads, modifying audio drivers, and writing low-level sound architecture code. Having these components running at the same time, and interacting with the hardware, was an amazing feeling."

  —Jeremy Bell, undergraduate

- "I learned a lot about DSP while working on the SP Cup, and since I am undertaking more DSP courses this semester, I feel more confident in my ability to understand more complicated concepts. I think my future career will almost certainly involve signal processing, so I will take the skills I have learned in DSP beyond university as well."

  —Jeremy Bell, undergraduate

- "I learned a lot about DSP algorithm design in general. I am also more confident in my understanding of sound architectures in Linux. I think I also learned a lot about teamwork, and what it takes to get things done under extreme time constraints."

  —Jeremy Bell, undergraduate

- "ICASSP was my first conference as an undergraduate, and I found it incredible. The amount of state-of-the-art technology and innovative creations was overwhelming, and it was almost impossible to keep up with in lectures. I was also surprised by the number of social events that occurred at the conference. It was great to be able to interact with so many talented and like-minded people on such a casual and friendly basis throughout the conference."

  —Jeremy Bell, undergraduate

- "We have already received several offers for other events at which we will be demonstrating the system. To do this will require some refinement of the interface and additional work on the software to make it more robust. Upon the graduation of our team, we will also be creating a handover document, so that future students can continue working on the system."

  —Team Beats on the Barbie

Team Madmom

- "I am interested in all topics making the link between music and mathematics, machine learning. I was working on incorporating online and real-time processing in the Madmom library when Sebastian told me that this work would be really useful for the SP Cup."

  —Amaury Durand, undergraduate

- "[Attending ICASSP was] really rewarding, it was my first time at a conference and, even though it was difficult for me to understand the talks I went to, I found it really interesting to meet the people who work on the topics that interest me."

  —Amaury Durand, undergraduate

- "[Participating in the SP Cup] was a perfect match. I just finished my Ph.D. in (mostly) offline beat and downbeat tracking, so it was very exciting for us to see how we can transform our system to work online and on an embedded device. Of course, it was more work than expected, but definitely a very exciting and rewarding experience!"

  —Florian Krebs, graduate mentor

- "The organization of everything was great, and I think there is no way to make this better. It was really great that you could organize a drum set, although this was not planned beforehand and not easy in a city that you don't know."

  —Florian Krebs, graduate mentor

- "It was very challenging given the limited processing power of the embedded device and extremely rewarding that it worked."

  —Sebastian Böck, supervisor.

Team Pulsebox

- "When I first heard of the topic for the 2017 SP Cup, I was very excited. As someone with a strong interest in both music theory and programming, I knew I had to get involved."

  —Creed Gallagher, undergraduate

- "One thing I learned a lot about was how to write truly speed-optimized code (Python with heavy use of NumPy). We had to push our Raspberry Pi to its limits. We also learned some lessons about the importance of effective communication and time management. We had to exercise a lot of discipline to complete such a big project on schedule."

  —Creed Gallagher, undergraduate

- "The signal processing challenge of beat tracking is incredibly complex! With so many types of songs and genres of music, there is no hard and fast rule as to what gives the best results. We ended up trying many approaches, many of which did not give as good results as we hoped. As a result, when we finally had something we felt performed well, it was incredibly satisfying."

  —Team Pulsebox

- "My senior project involves continual development of the PulseBox. I want to eventually create a 3-D holographic display that tracks both the beat and 'mood' of a song. Sebastian's team convinced us that the future of musical analysis lies in the use of neural networks, which is the avenue I will be exploring."

  —Creed Gallagher, undergraduate

- "As an undergraduate, attending ICASSP was an amazing and humbling experience. I enjoyed listening in on the presentations which gave me a window into the cutting edge of SP research. Plus, everyone was friendly and New Orleans was a fun venue."

  —Creed Gallagher, undergraduate

Forthcoming project competitions for undergraduates

The fifth edition of the SP Cup will be held at ICASSP 2018. The theme of the 2018 competition will be announced in September. Teams who are interested in the SP Cup competition may visit this link: https://signalprocessingsociety.org/get-involved/signal-processing-cup.

In addition to the SP Cup, the IEEE SPS recently announced the first edition of the Video and Image Processing Cup. The final competition will be held at the IEEE IEEE International Conference on Image Processing, in Beijing, China, 17–20 September. The theme of this competition is "Challenging Road Sign Detection." For details, visit: https://signalprocessingsociety.org/get-involved/video-image-processing-cup.

Acknowledgments

As the SP Cup 2017 Organizing Committee, we would like to express our gratitude to all of the people who made this adventure a reality: the participating teams, the judging panel, the local organizers, the IEEE SPS Membership Board for its financial support for the drum kit rental, and MathWorks for its sponsorship. Matthew E.P. Davies is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the project IF/01566/2015.

Authors

Craig T. Jin (craig.jin@sydney.edu.au) is an associate professor at the University of Sydney, Australia. He is a Senior Member of the IEEE and a current member of the IEEE Audio and Acoustic Signal Processing Technical Committee (AASP TC). He initiated this edition of the SP Cup on behalf of the AASP TC, developed the competition specification and pedagogical materials, and ran the competition alongside Matthew Davies.

(continued on page 170)

Exposition, Atlanta, GA, June 2013, pp. 23.828.1–23.828.21.

[8] A Guide to the flipped classroom. The Chronicle of Higher Education (2015, Jan. 07) [Online]. Available: http://www.chronicle.com/article/A-Guide-to-the-Flipped/151039/

[9] J. L. Jensen, T. A. Kummer, and P. D. D. M. Godoy, "Improvements from a flipped classroom may simply be the fruits of active learning," *CBE Life Sci. Educ.*, vol. 14, no. 1, pp. 1–12, Mar. 2015.

[10] E. F. Gehringer, "Resources for "flipping" classes," in *Proc. ASEE Annu. Conf. Exposition*, Seattle, WA, June 2015, pp. 26.1336.1–26.1336.10.

[11] J. O'Flaherty and C. Phillips, "The use of flipped classrooms in higher education: A scoping review," *Internet Higher Educat.*, vol. 25, pp. 85–95, Oct. 2015.

[12] Flip Learning: Research, Reports, and Studies. Flipped Learning Network. [Online]. Available: http://flippedlearning.org/research-reports-studies/

[13] B. J. Limbach and W. L. Waugh, "Questioning the lecture format," *NEA Higher Educ. J. Thought Action*, vol. 20, no. 1, pp. 47–56, 2005.

[14] M. Freeman, P. Blayney, and P. Ginns, "Anonymity and in class learning: The case for electronic response systems," *Australasian J. Educ. Tech*, vol. 22, no. 4, pp. 568–580, 2006.

[15] J. R. Stowell and J. M. Nelson, "Benefits of electronic audience response systems on student participation, learning, and emotion," *Teaching Psychol.*, vol. 34, no. 4, pp. 253–258, 2007.

[16] C. R. Graham, T. R. Tripp, L. Seawright, and G. Joeckel, "Empowering or compelling reluctant participators using audience response systems," *Active Learn. Higher Educ.*, vol. 8, no. 3, pp. 233–258, 2007.

[17] Y. K. Kim and L. J. Sax, "Student–faculty interaction in research universities: Differences by student gender, race, social class, and first-generation status," *Res. Higher Educ.*, vol. 50, no. 5, pp. 437–459, 2009.

[18] W. Griffin, S. D. Cohen, R. Berndtson, K. M. Burson, K. M. Camper, Y. Chen, and M. A. Smith, "Starting the conversation: An exploratory study of factors that influence student office hour use," *College Teach.*, vol. 62, no. 3, pp. 94–99, 2014.

[19] P. C. Blumenfeld, E. Soloway, R. W. Marx, J. S. Krajcik, M. Guzdial, and A. Palincsar, "Motivating project-based learning: Sustaining the doing, supporting the learning," *Educ. Psychol.*, vol. 26, no. 3-4, pp. 369–398, June 1991.

[20] H. A. Hadim and S. K. Esche, "Enhancing the engineering curriculum through project-based learning," in *Proc. 32nd Annu. Frontiers in Education (FIE'02)*, Nov. 2002, vol. 2, pp. F3F.1–F3F.6.

[21] M. Frank, I. Lavy, and D. Elata, "Implementing the project-based learning approach in an academic engineering course," *Int. J. Tech. Design Educat.*, vol. 13, no. 3, pp. 273–288, Oct. 2003.

[22] J. S. Krajcik and P. C. Blumenfeld, "Project-based learning," in *The Cambridge Handbook of the Learning Sciences*, R. K. Sawyer, Ed. New York, NY: Cambridge Univ. Press, 2006, ch. 19, pp. 317–334.

[23] J. Bourne, D. Harris, and F. Mayadas, "Online engineering education: Learning anywhere, anytime," *J. Eng. Educ.*, vol. 94, no. 1, pp. 131–146, Jan. 2005.

[24] L. Pappano. (Nov. 2, 2012). The year of the MOOC. The New York Times. [Online]. Available: http://www.nytimes.com/2012/11/04/education/edlife/massive-open-online-courses-are-multiplying-at-a-rapid-pace.html

[25] D. F. O. Onah, J. Sinclair, and R. Boyatt, "Dropout rates of massive open online courses: Behavioural patterns," in *Proc. 6th Intl. Conf. Education and New Learning Technologies (EDULEARN'14)*, Barcelona, Spain, July 2014.

[26] T. A. Baran, R. G. Baraniuk, A. V. Oppenheim, P. Prandoni, and M. Vetterli, "MOOC adventures in signal processing: Bringing DSP to the era of massive open online courses," *IEEE Signal Process. Mag.*, vol. 33, no. 4, pp. 62–83, July 2016.

[27] Khan Academy. [Online]. Available: http://www.khanacademy.org/

[28] R. H. Rockland, L. Hirsch, L. Burr-Alexander, J. D. Carpinelli, and H. S. Kimmel, "Learning outside the classroom—Flipping an undergraduate circuits analysis course," in *Proc. ASEE Annu. Conf. Exposition*, Atlanta, GA, June 2013, pp. 23.854.1–23.854.8.

[29] B. Van Veen, "Flipping signal-processing instruction," *IEEE Signal Process. Mag.*, vol. 30, no. 6, pp. 145–150, Nov. 2013.

[30] M. L. Fowler, "Flipping signals and systems—Course structure & results," in *Proc. IEEE Intl. Conf. Acoustics, Speech, and Signal Processing (ICASSP'14)*, Florence, Italy, May 2014, pp. 2219–2223.

[31] G. J. Kim, M. E. Law, and J. G. Harris, "Lessons learned from two years of flipping Circuits I," in *Proc. ASEE Annu. Conf. Exposition*, Seattle, WA, June 2015, pp. 26.1087.1–26.1087.12.

[32] M. G. Schrlau, R. J. Stevens, and S. Schley, "Flipping core courses in the undergraduate mechanical engineering curriculum: Heat transfer," *Adv. Eng. Educ.*, vol. 5, no. 3, Nov. 2016.

[33] J. R. Buck, K. E. Wage, and J. K. Nelson, "Designing active learning environments," *Acoustics Today*, vol. 12, no. 2, pp. 12–20, 2016.

[34] W. U. Bajwa. SigProcessing YouTube channel. [Online]. Available: http://www.youtube.com/user/SigProcessing

[35] Poll Everywhere. [Online]. Available: http://www.polleverywhere.com/

[36] R. A. Layton, M. L. Loughry, M. W. Ohland, and G. D. Ricco, "Design and validation of a web-based system for assigning members to teams using instructor-specified criteria," *Adv. Eng. Educat.*, vol. 2, no. 1, pp. 1–28, 2010.

[37] CATME System. [Online]. Available: http://info.catme.org/

SP

SP COMPETITIONS *(continued from page 150)*

Matthew E.P. Davies (matthew.davies@inesctec.pt) is a senior researcher at INESC TEC, Portugal. He is an IEEE Member and an associate editor for *IEEE/ACM Transactions on Audio, Speech, and Language Processing*. He is responsible for the technical components of the competition and its materials and ran the competition alongside Craig T. Jin.

Patrizio Campisi (patrizio.campisi@uniroma3.it) is a professor at Roma Tre University, Rome, Italy. He chairs the Student Service Committee of the IEEE Signal Processing Society and served as the general chair of the 2015 Information Forensics and Security Technical Committee. He is the organizer for the SP Cup since the 2015 edition and the initiator of the Video and Image Processing Cup.

References

[1] M. Goto and Y. Muraoka, "A beat tracking system for acoustic signals of music," in *Proc. 2nd ACM Int. Conf. Multimedia*, 1994, pp. 365–372.

[2] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *J. Acoust. Soc. Am.*, vol. 103, no. 1, pp. 588–601, 1998.

[3] F. Gouyon and S. Dixon, "A review of automatic rhythm description systems," *Comput. Music J.*, vol. 29, no. 1, pp. 34–54, 2005.

[4] X. Serra, M. Magas, E. Benetos, M. Chudy, S. Dixon, A. Flexer, E. Gómez, F. Gouyon, P. Herrera, S. Jordà, O. Paytuvi, G. Peeters, J. Schlüter, H. Vinet, and G. Widmer. (2013). Roadmap for Music Information ReSearch. London: MIRES Consortium. [Online]. Available: http://mires.eecs.qmul.ac.uk/about.html

[5] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," Queen Mary Univ., Centre for Digital Music, London, Tech. Rep. C4DM-TR-09-06, 2009.

[6] J. Oliveira, M. E. P. Davies, F. Gouyon, and L. P. Reis, "Beat tracking for multiple applications: A multi-agent system architecture with state recovery," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 20, no. 10, pp. 2696–2706, 2012.

[7] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. "Madmom: A new Python audio and music signal processing library," in *Proc. 2016 ACM Multimedia Conf.*, 2016, pp. 1174–1178.

SP

JOURNAL OF NEW MUSIC RESEARCH, 2018
VOL. 47, NO. 2, 114–128
https://doi.org/10.1080/09298215.2017.1409769

Routledge
Taylor & Francis Group

A generative model for the characterization of musical rhythms

George Sioros[a], Matthew E. P. Davies[b] and Carlos Guedes[a,b]

[a]INESC-TEC, Sound and Music Computing Group, Porto, Portugal; [b]Arts & Humanities, New York University Abu Dhabi, Abu Dhabi, United Arab Emirates

ABSTRACT

We present a novel model for the characterization of musical rhythms that is based on the pervasive rhythmic phenomenon of syncopation. Syncopation is felt when the sensation of the regular beat or pulse in the music is momentarily disrupted; the feeling arises from breaking more expected patterns such as pickups (anacrusis) and faster events that introduce and bridge the notes articulated on the beats. Our model begins with a simple pattern that articulates a beat consistent with the metrical expectations of a listener. Any rhythm is then generated from a unique combination of transformations applied on that simple pattern. Each transformation introduces notes in off-beat positions as one of three basic characteristic elements: (1) syncopations, (2) pickup rhythmic figures and (3) faster notes that articulate a subdivision of the beat. The characterization of a pattern is based on an algorithm that discovers and reverses the transformations in a stepwise manner. We formalize the above transformations and present the characterization algorithm, and then demonstrate and discuss the model through the analysis of the main rhythmic pattern of the song 'Don't stop 'till you get enough' by Michael Jackson.

ARTICLE HISTORY
Received 31 March 2017
Accepted 14 November 2017

KEYWORDS
Rhythm; music analysis; metre; syncopation; transformation; timing

1. Introduction

Many musical rhythms elicit a sense of periodicity and regularity in listeners. Simple or complex music, even when it is not repetitive, often evokes the sensation of a regular pulse (Parncutt, 1987, 1994) which is evident when we tap in synchrony with music. These pulses feature accents where some are perceived as stronger than others. These structured, accented pulses form our expectations about the timing of the rhythmic events and are at the foundation of musical metre (Honing, 2012; Jones, 2008; Jones, Moynihan, MacKenzie, & Puente, 2002; McAuley, 2010, p. 168).

Besides metre, the perception of musical rhythm involves another mechanism: the grouping of events, referred to also as serial grouping or 'figural coding' (Parncutt, 1994, p. 412). While the strong and weak pulses of metre group non-adjacent events in a periodic way, figural coding groups adjacent events primarily based on their proximity in time. When we listen to a rhythm, strong interactions between these mechanisms and the pattern of physical durations occur in our mind as we interpret the rhythm by inferring a metrical context (Clarke, 1987a, 1987b; Fraisse, 1982; Honing, 2012). Manifestations of the interaction between our metrical expectations and the pattern of durations are the feeling of syncopation (Huron, 2006, p. 295) and anacrusis or 'pickup' (Lerdahl & Jackendoff, 1981, p. 500).

According to Huron (2006, p. 295), an event in a relatively weak metrical pulse (off-beat) creates a higher expectation for an event in the following stronger pulse (on-beat). Our expectation can either be confirmed by a following event resulting in the proper binding of the two (London, 2012, p. 107), or be violated when the expected event does not occur, in which case a syncopation is felt (see also Huron & Ommen, 2006). In this sense, an event in a weak position is bonded to the following event in the strong position, or it syncopates when this bond is broken.

In this article, we present a model for the systematic analysis and characterization of musical rhythms based on Huron's (2006, p. 295) metrical expectation principle of weak–strong bonds described above. The model identifies the characteristic relations of music events to a given beat, by codifying their deviations from the on-beat positions as: (1) syncopations, (2) anacrusis or pickups or (3) the mere articulation of a metrical subdivision.

Sioros and Guedes (2014) generate and analyse syncopation by anticipating events in weaker metrical positions. Such shifts of events were introduced in a qualitative manner by Temperley in his study of syncopation in rock

CONTACT George Sioros georgios.sioros@fe.up.pt INESC-TEC, Sound and Music Computing Group, Porto, Portugal

music (1999). Here, we extend both approaches to include two more transformations besides the syncopation shifts: (1) figural shifts that generate anacruses (pickups) and (2) density transformations that insert new events. The three transformations together form a complete model for the characterization of a rhythmic pattern, in a manner similar to a Schenkerian analysis. The model is able to generate any rhythm starting from a metronome-like pattern articulating only beat positions. Each transformation brings a particular element to the rhythm. Thus, characterizing the pattern centres on the discovery of the unique set of transformations that generates it.

Our model is primarily motivated by its analytical applications in computer algorithms. The approach to rhythm analysis through transformations is well established. Several models are based on abstract mathematical transformations (Louboutin & Meredith, 2016; Meredith, 2014; Paiement, Grandvalet, Bengio, & Eck, 2007) or properties particular to a certain class of rhythms (G. Toussaint, 2002; Toussaint, 2013). Most noticeably, Lewin (2007) formalized generalized algebraic transformations that can be applied to various musical dimensions including the timing of events. Although, the transformations we present here could be expressed as Lewin's generalized transformations, we do not pursue an integration of our model into Lewin's theory. Our approach is different. We formulate transformational definitions for music phenomena, such as syncopation, based on theories and cognitive principles of rhythm and metre perception. In this way, we aim to create a model that reflects a listener's interpretation of a rhythm in a metrical context and, thus, provide a musically meaningful codification of the patterns.

We formalize the transformations as generative processes that introduce new elements to a rhythm but we do not explicitly define their inverse. Instead, our focus is in the characterization method that we developed which effectively reverses the transformations. For instance, while we define a transformation that introduces new syncopation to a pattern, we do not define an inverse de-syncopating transformation. Instead, all instances of syncopation are identified and the respective de-syncopated patterns are generated during the characterization of the pattern.

The characterization provided by the model is unique and unambiguous with respect to a regular beat upon which weak–strong bonds of the musical events are expected. Yet, the model does not exclude the interpretation of the same pattern according to a different beat. For example, a pattern might be understood as syncopated or not, depending on the placement of the beat duration relative to the pattern. Different listeners may infer different metrical contexts to interpret a rhythm (London, 2012), depending on a variety of factors, such as cultural background, music style, or other elements of the music besides the timing of the events. Often, the metre itself is intentionally ambiguous, so that such metrical ambiguity will be inherent in the characterization. Our systematic approach to rhythm analysis does not imply a unique interpretation based on the 'correct' metre of the music. In fact, the proposed model can serve as the basis for comparisons of different interpretations of a single rhythm that are based on different metrical expectations.

The remainder of the paper is structured as follows. In Section 2, we present the transformations and the characterization algorithm upon which the rhythm model is based. In Section 3, we demonstrate the model in practice using the example of the Michael Jackson song 'Don't stop 'till you get enough'. Finally, in Section 4, we discuss important aspects of the model and of its similarities and differences to other relevant approaches.

2. Rhythm generation model

The rhythm generation model consists of a set of three generative transformations that, given the beat, can generate any rhythmic pattern and a characterization algorithm that identifies the previously applied transformations and reverses them. In Subsection 2.1, we formalize the transformations and define the transformation vector, i.e. a consistent way of representing them. Subsequently, in Section 2.2, we present the characterization algorithm. For the algorithm to effectively characterize a rhythm, the transformations need to be reversible in a unique and unambiguous way. To this end, we impose two reversibility constraints that we discuss in the final Subsection 2.3.

The transformations are derived from the definition of syncopation as a cognitive mechanism related to metrical expectations and the weak–strong bonds between events articulated in weak–strong metrical positions (Huron, 2006, p. 295). Consequently, their formalization first requires a formalized representation of metrical expectations, i.e. a metrical template. We base our metrical templates on the stratification of metre by Yeston (1976) and on the similar definition of metre by Lerdahl and Jackendoff (1983). We use the algorithm devised by Barlow and Lohner (Barlow, 1987; Barlow & Lohner, 1987), which was adapted by Sioros and Guedes (2014) for the purpose of defining syncopation, to automatically generate the template of any given metre. A metrical template consists of hierarchically grouped pulses[1] as in the example of Figure 1.

A lower and upper threshold for the duration of the metrical subdivisions included in the template is set based

[1]In the examples of this article, we follow the convention of numbering the pulses and metrical subdivisions of a template starting at 0.

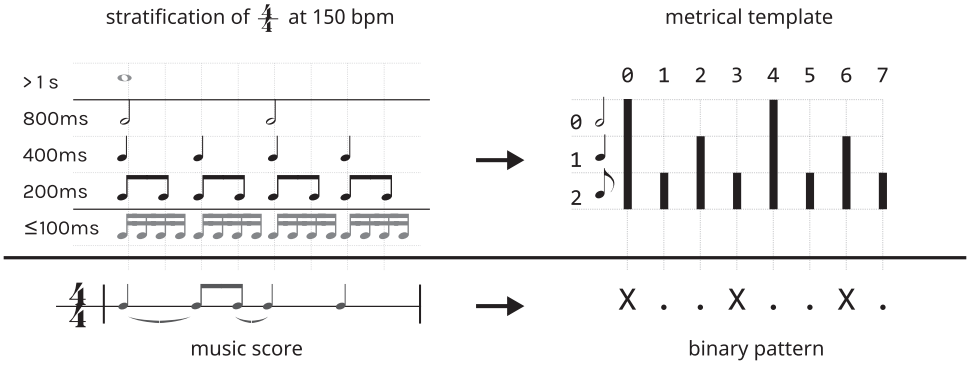


Figure 1. Example of metrical templates and binary representation of rhythms. Left: A rhythmic pattern in 4/4 at 150 bpm. Above the pattern, the **metre** is stratified to its metrical subdivisions. Right: The binary representation of the pattern (X = event, . = silent pulse). Above it, the metrical template is shown according to the stratification of the **metre** on the left. Metrical levels slower than 1s or faster than 100 ms are ignored as described in detail by Sioros and Guedes (2014); London (2012, Chapter 2).

on the metrical salience of the subdivisions (Parncutt, 1994). The lower threshold has been estimated in numerous studies to be around 100 ms since faster durations are not perceived as metrical subdivisions (London, 2012, p. 29; Parncutt, 1994; Repp, 2006). The upper threshold for the template is set to the most salient subdivision, which we will hereafter call the beat, and is estimated around 1s (Parncutt, 1994). The prominent role given to the beat in our model follows from the weakening of the syncopation feel and of the weak–strong bond for slow metrical levels as discussed in detail by Sioros and Guedes (2014). In the example of Figure 1, the sixteenth note and the bar levels are omitted as they fall outside of the above thresholds.

The characterization of rhythms presented here is only based on the quantized onsets of the musical events. As each onset in the rhythm belongs to a metrical position, the metrical template reduces a rhythm into a binary string. In the examples of this section we use binary rhythm representations. However, in the more musical example of Section 3, the events have other properties besides their timing, such as pitch or lyrics. The transformations and characterization process preserve these properties.

The model consists of three main transformations: (1) the syncopation transformations, which we denote with the letter 'S', that place events in weak metrical positions that are not properly bound to following events in strong positions and therefore syncopate (Sioros & Guedes, 2014), (2) the figural transformations, which we denote with 'F', that result in the binding of events in weak–strong rhythmic figures of an anacrustic character and (3) density transformations, which we denote with 'D', that insert events in weak metrical positions surrounded by events in stronger pulses.

Examples of the transformations are shown in Figure 2. Accordingly, syncopation is the result of the anticipation of an event from a strong metrical position to a preceding weaker one. In a similar manner, a weak–strong rhythmic figure is generated by delaying an event from a strong metrical position to a following weaker one, in order for it to approach a later stronger event. Density transformations do not involve the shifting of events. Instead, a new event is inserted between two existing events so that it articulates a faster metrical level.

In the framework of this model, we call a metronome any pattern that articulates only beat positions i.e. that comprises events only at the slowest metrical level included in the metrical template. Departing from a metronome, the transformations can generate any binary rhythmic pattern in a step-wise process, introducing a single new element of the rhythm with each step—either a syncopation, an anacrusis or a new event. Conversely, given the metrical expectations that derive from a metronome, one can characterize a rhythmic pattern by identifying and reversing those transformations.

2.1. Transformation vectors

Each elementary transformation is characterized by the metrical level difference between the two positions comprising a weak–strong bond. In the case of syncopation, even though the bond is not realized, it is expected. For example, the syncopations of Figure 2(a) on the half note level (pulse 8) are produced by the previous quarter (pulse 4), eighth (6) or sixteenth (7) notes. In fact, the difference of the metrical levels has been used as a means to quantify the feeling of syncopation (Longuet-Higgins & Lee, 1984). The anacruses of the figural transformations in Figure 2(b) are characterized by the same level differences; an event in the quarter, eighth or sixteenth notes (pulses 4, 6 or 7) preceding an event in the half note position (pulse 8). Similarly, the inserted events of the

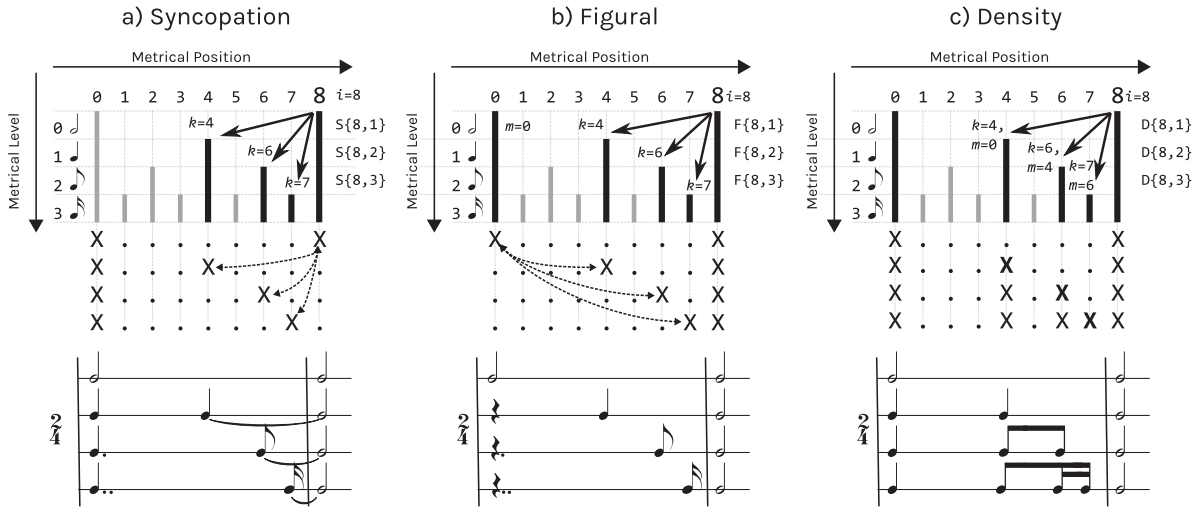


Figure 2. Definition of Syncopation, Figural and Density transformations. Each transformation corresponds to a vector (solid arrows) consisting of the metrical position on which the transformation is applied (pulse 8) and the metrical level difference to preceding weaker pulse. The syncopation and figural vectors point to the pulse an event is shifted to, while the density vectors point to the pulse on which a new event is inserted. The dotted arrows represent the shift of an event.

density transformations in the last column of Figure 2(c) are placed a quarter, eighth or sixteenth note (pulses 4, 6 or 7) before the following half note (in pulse 8).

In general, given a metrical template, any of the generative transformations presented here, can be completely and uniquely defined by two components that together constitute the transformation vector: (1) a relatively strong pulse on which the transformation is applied and (2) the metrical level difference from a preceding weaker pulse that can form a weak–strong bond. A weak–strong bond always consists of consecutive metrical positions, that is, all pulses between them must belong to weaker metrical positions (faster metrical levels); otherwise they would not correspond to the expected proper binding of events (London, 2012, p. 107). If we assign indices and metrical levels to the pulses as in the Figure 2, then a vector can be formed as: {pulse index *i*, metrical level difference *d*}. The symbol *S*, *F* or *D* should be used in front of the vector to distinguish between syncopation, figural and density transformations, respectively.

2.1.1. Syncopation

Generating the syncopation described in the vector *S* {*i, d*} consists of finding the first pulse *k* preceding pulse *i* that belongs to the weaker metrical level *l(k) = l(i)-d* and shifting (anticipating) the event from pulse *i* to pulse *k* (Sioros & Guedes, 2014). In the example in Figure 2(a), the syncopation *S* {8, 1} is generated by finding the pulse that belongs one metrical level faster than pulse 8 and shifting the event of pulse 8 to that position. Since pulse 8 belongs to the half note metrical level, the event is shifted to the previous quarter note position. In the *S* {8, 2} example, the same event is shifted to the previous eighth note position.

2.1.2. Figural

Generating the weak–strong rhythmic figure described in the vector *F* {*i, d*} consists in finding the first pulse *k* preceding pulse *i* that belongs to the weaker metrical level *l(k) = l(i)-d*, in the same manner as in the syncopation transformation. However, instead of anticipating the event of pulse *i*, we delay the event preceding pulse *k*. The transformation takes place only if the event is shifted to a pulse weaker than the one it initially belonged to. In the *F* {8, 1} example of Figure 2(b), pulse *k* is the previous quarter note position (4) and the event preceding *k* is found at the previous downbeat (0). The event is therefore delayed from pulse 0 to 4. Similarly, in the *F* {8, 2} example, the event is delayed to the eighth-note position preceding the second bar. In this way, the figural transformations create a weak–strong rhythmic figure with the event in the following beat.

2.1.3. Density

The density transformation *D* {*i, d*} inserts a new event in pulse *k* preceding an event in pulse *i*. Pulse *k* is determined by the metrical level difference *d* as in the other two transformations. However, the event is inserted under two conditions: (1) the event preceding pulse *k* must belong to a stronger pulse *m*, and (2) pulse *k* must be the strongest pulse between pulses *i* and *m*. These two conditions ensure that the inserted event will not constitute a simultaneous syncopation or figural transformation; an essential requirement for the characterization algorithm and the reversibility of the transformations as discussed in Sections 2.2 and 2.3.

Two important remarks should be made. First, the vector *S* {*i, d*} in syncopation transformations coincides with

the actual shift of an event from pulse *i* to a preceding weaker position. However, in figural transformations, the shifted event is not found in the vector but it is obtained indirectly as the previous event from the weak position determined by the vector. In the example of Figure 2, the solid arrows representing the vectors and dotted arrows representing the shifts coincide for syncopation but not for the figural transformations. Regardless, the way the transformations and vectors are formalized ensures that each vector always describes a unique shift.

The second remark is related to the special case of ternary subdivisions of the beat as in the example in Figure 3. By definition, the transformation vectors refer to the relation of two positions of different metrical strength. While the two ternary subdivisions of a beat belong to the same weak metrical level, Palmer and Krumhansl provide evidence that there is a difference in their metrical strength. They found that the frequency of musical events in the different metrical positions (Palmer & Krumhansl, 1990; Figure 1) coincide with the theoretical strength of the corresponding positions for common time signatures. However, musical events were significantly more probable in the third and sixth-eighth note positions of a 6/8 metre than in the second and fifth positions. In other words, the second of the two ternary subdivisions of the beat was found to be metrically stronger than the first even though they both belong to the same metrical level. Furthermore, their results agree with goodness-of-fit judgments of temporal patterns (Palmer & Krumhansl, 1990; Figure 3) and with theoretical calculations (Barlow, 1987; Barlow & Lohner, 1987).

This finding suggests that a note articulated in the first of the two positions of a ternary subdivision of the beat is normally followed by a note in the subsequent second weak position, as in pattern E, forming a weak–weak rhythmic figure. Therefore, such a weak–weak rhythmic binding of events is of the same nature as the weak–strong binding of events discussed by Huron (2006) and London (2012, p. 107) so that it can be generated by a figural (pattern C, transformation *F* {4, 0}) or density (pattern D, *D* {4, 0}) transformation and that in the absence of the second note a syncopation is felt (pattern B, transformation *S* {4, 0}).

In addition, Povel and Okkerman (1981) argue that a rhythmic pattern as in Figure 3(B)—although it does not constitute a typical example of syncopation—is 'creating a special rhythmical tension' because of the ambivalence between the metrical accents and the accent caused by the short–long durations. In contrast, a long-short figure (as in Figure 3(A)) does not create any tension and agrees with the metre. The evidence from Palmer and Krumhansl (1990) and the syncopation transformations presented here suggest that this 'special tension' is of the same nature as the more typical examples of syncopation with the difference that it is, perhaps, felt less strongly.

2.2. Characterization algorithm

The characterization of a rhythm is based on identifying and reversing any previously applied transformations, one by one, by shifting or removing the appropriate events. By identifying, we mean determining in a unique way the kind of transformation—whether it describes a syncopation, figural or density—and the corresponding vector. The process of identifying and reversing a transformation on pulse *i* of a given pattern is shown in the flow diagram of Figure 4.

The first step in the algorithm (marked 1 in the flow chart) determines the second component of the vector that corresponds to a potential transformation. The weak–strong pair of pulses *k* and *i* must comply with the definition of the transformation vector so that all pulses between them must belong to faster metrical levels. If a valid weak event is found, then the algorithm proceeds in determining whether the vector corresponds to a syncopation,

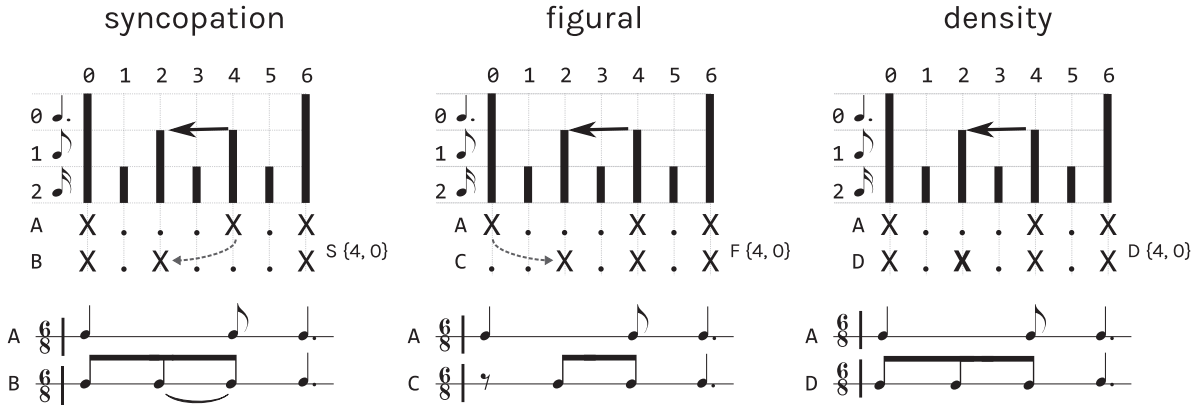


Figure 3. Example of vectors with 0 metrical level difference.

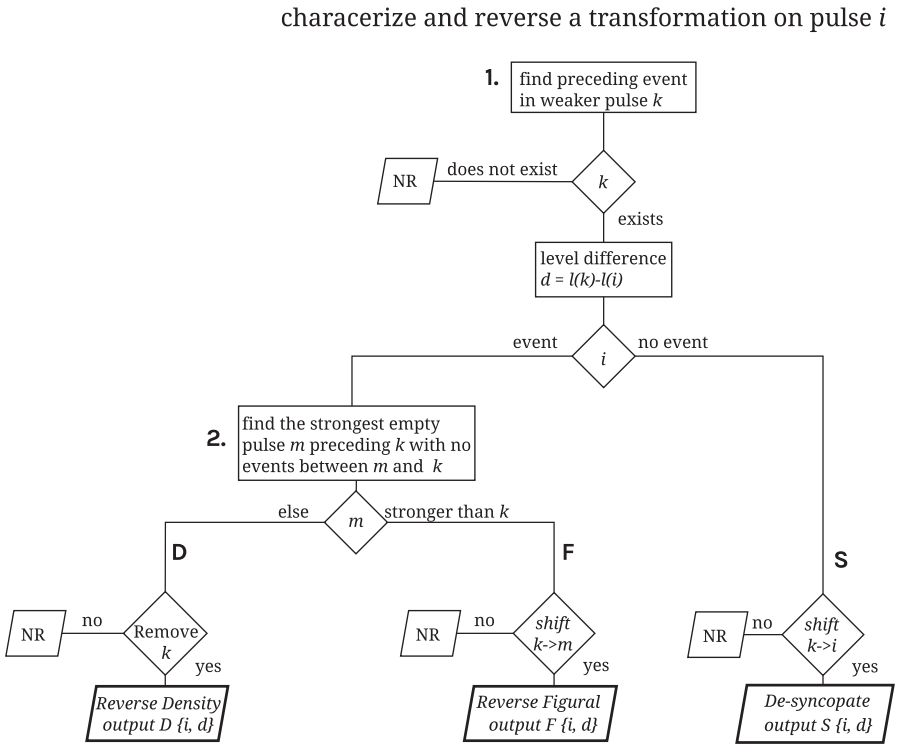


Figure 4. Flow diagram of the process of characterizing and reversing a single transformation previously applied on pulse *i*. The processes and decisions made are subject to the reversibility constraints described in the Section 2.3. The NR outputs indicate that no transformation is reversed because either no transformation was found (step 1) or a transformation on another pulse must be reversed first (step 2).

figural or density transformation. In the absence of an event in pulse *i*, a syncopation is identified.

In the opposite case, the algorithm distinguishes between a potential figural and density transformation. It looks for the strong pulse *m* that precedes the weak event and which would be its original position (marked 2 in the flow chart). If no such valid 'origin' for the event is found, then a density transformation is identified.

The last step in all the above cases is the reversing of the identified transformation (marked *S*, *F* or *D* in the flow chart) and the output of the corresponding vector. This step is restricted by the reversibility constraints described in Section 2.3. If reversing is not possible the algorithm outputs *NR* signifying the identified transformation cannot be reversed without first reversing a transformation on another pulse.

In our model, it is possible for a single pulse to undergo more than one transformation. All transformations in the pattern will be identified and reversed by applying the above algorithm recursively to all the pulses until no transformation is found.

2.3. Reversibility constraints

Reversibility is essential and, in many ways, shaped how the transformations are formalized, whereby any elementary transformation should always be reversible in a unique way. This ensures that, given a metrical template, any rhythm can be uniquely represented by a series of transformations. Conversely, it guarantees that a series of transformations can only generate a single pattern when applied to a metronome. Reversing a transformation is essentially a process of characterizing it. For example, de-syncopating is equivalent to characterizing the syncopation that is to be reversed. Having more than one way to reverse a transformation would mean that there is more than one way to characterize it, and thus the analysis of the pattern would be subject to interpretation. Therefore, the reversibility rule makes for a consistent model.

We impose two constraints on the transformations presented above to ensure that the above reversibility requirement is always met:

(1) The order of events must always be preserved.
(2) Each shift must correspond to a single transformation and a single vector.

In the following, we discuss the implications of the above constraints using appropriate examples.

The first constraint, which requires that for a shift of an event to be possible between two pulses, all in between pulses must be silent (i.e. not carrying an event) is discussed in detail by Sioros and Guedes (2014). In Figure 5,

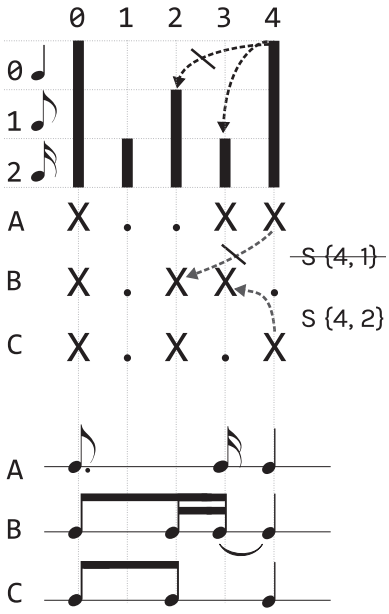


Figure 5. Example of the first reversibility constraint. The strikethrough transformation *S* {4, 1} is forbidden since it does not respect the initial order of events. Pattern B is generated through the *S* {4, 2} transformation.

we provide an example of how reversibility is broken if this constraint is not adhered to. If we imagine the transformation *S* {4, 1} on pattern A of the figure, the event of pulse 4 would be shifted to pulse 2 over the event in pulse 3. However, pattern B can be the result of the *S* {4, 2} transformation on pattern C. The reversibility constraint requires that pattern C be the result of only one of the two transformations. As the syncopation found in pattern B is de-syncopated to pattern C, the *S* {4, 1} transformation must be forbidden, else it would break the reversibility constraint.

The second constraint dictates that generating or reversing a transformation should neither generate nor reverse another transformation simultaneously. If such a step were allowed, then the transformations would not correspond to a single vector. It is precisely this constraint which ensures that each transformation introduces a single distinct element in the rhythm. The conditions under which the figural and density transformations can take place are the result of this constraint. We demonstrate the way it is applied and discuss some of its implications in the context of the two examples presented in Figures 6 and 7.

Figure 6 is an example of a figural transformation that would simultaneously generate a syncopation. If the *F* {4, 2} transformation were allowed, the weak–strong bond between the events in pulses 2 and 3 would be broken. In our model, such improper binding can only be the result of a syncopation transformation, which in this example could only be the transformation *S* {2, 1} applied on pattern C. In contrast, a weak–strong bond between two events can arise from either a figural or a density transformation. In pattern B, the bond between the two last events can be generated through the density transformation *D* {4, 2} instead of a figural prior to introducing the syncopation *S* {2, 1}. In this way, each element in the rhythm corresponds to a unique weak–strong bond. When trying to characterize a potential figural transformation, the constraint takes an equivalent form: reversing a figural transformation is not allowed if it simultaneously removes an existing syncopation.

The second example, which is shown in Figure 7, refers to the potential masking of syncopations. In this case, the second constraint dictates that de-syncopating one instance of syncopation should not generate another, or in its reverse form, that generating a syncopation should not de-syncopate another syncopation. Pattern C contains a syncopation (felt on pulse 4) that could be de-syncopated as shown in the right side of the figure. However, this would result in the generation of a syncopation on pulse 2 (pattern E). Instead, we should first reverse the figural transformation (pattern B) and then reverse the syncopation on pulse 4, leading to pattern A. Conversely, the syncopation *S* {4, 1} cannot be generated on pattern E as it would mask the existing syncopation *S* {2, 1}.

The last example addressed the issue of applying a series of transformations to a pattern. We will demonstrate the process of characterizing a rhythm that has undergone a series of transformations using the example of Figure 8. In the figure, besides the detailed transformation vectors shown on the binary representation of the patterns, a simplified representation of the vectors is shown on the staves above the corresponding metrical positions. In this representation, the symbol depicts the kind of transformation (*F*: figural, *S*: syncopation, *D*: density) and the number the metrical level difference (e.g. *F* {4, 1} - > *F*1, or *D* {4, 2} - > *D*2). The pulse index is omitted as it is implied by the placement of the vector at the corresponding metrical position.

Starting with pattern A in Figure 8, we scan the pulses from left to right and reverse the transformations following the process detailed in Section 2.2:

- Pulses 0 to 3 have not undergone any transformations since they are not preceded by events in weaker metrical positions.
- Pulse 4 carries an event which is preceded by an event in the weaker pulse 3. As the preceding stronger pulse 2 is not empty, a density transformation is identified and reversed.
- The next pulse preceded by a weak event is pulse 10. In this case, the weak event in pulse 9 is preceded by an empty stronger position (8) so that the event

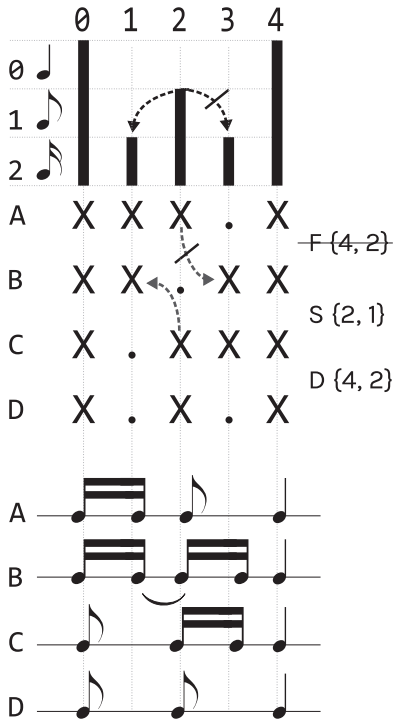


Figure 6. The strikethrough transformation *F* {4, 2} violates the second constraint. It simultaneously generates the weak–strong rhythmic figure between pulses 3 and 4 and the syncopation *S* {2, 1} between pulses 1 and 2.

is shifted there reversing the figural transformation *F* {10, 1}.

- Next, a syncopation is found in pulse 12: the silent pulse 12 is preceded by the event in the weaker pulse 10. The syncopating event is shifted to pulse 12 yielding the vector *S* {12, 1}. This shift is only possible because the figural transformation *F* {10, 1} has already been reversed in the previous step. Had it been otherwise, reversing the syncopation would violate the second constraint as it would simultaneously generate a new syncopation in pulse 10.
- Continuing in the same fashion to the following pulses, the syncopation *S* {16, 2} is identified and reversed, followed by the density transformation *D* {20, 1}.
- Starting the process again from the beginning, pulse 4 is now preceded by the weaker event in pulse 2. This time, the transformation is identified as figural because of the lack of an event on the previous stronger beat (pulse 0).
- Reaching pulse 16, the weak event between the fourth and fifth beat is removed yielding the density transformation *D* {16, 1}.
- Now all events in the pattern belong to beat positions and the algorithm cannot identify any further transformations. The characterization of the pattern is therefore complete.

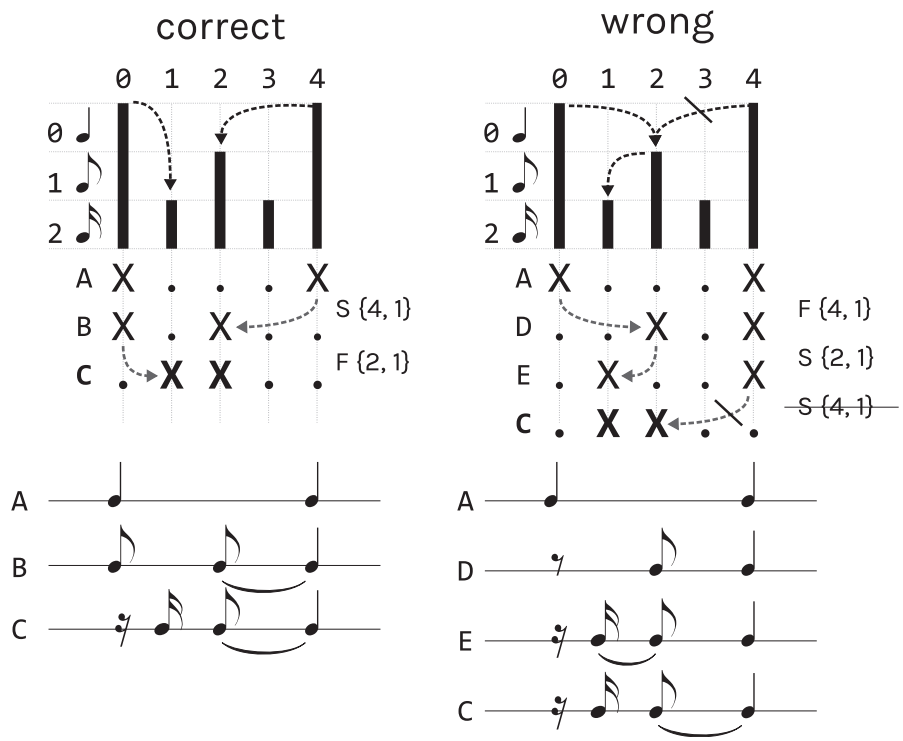


Figure 7. Example of the second reversibility constraint. The bottom pattern is the result of the left-side (correct) transformations rather than the right-side ones (wrong). Introducing the syncopation *S* {4, 1} on pattern E 'masks' the previous syncopation *S* {2, 1} and therefore is not allowed.

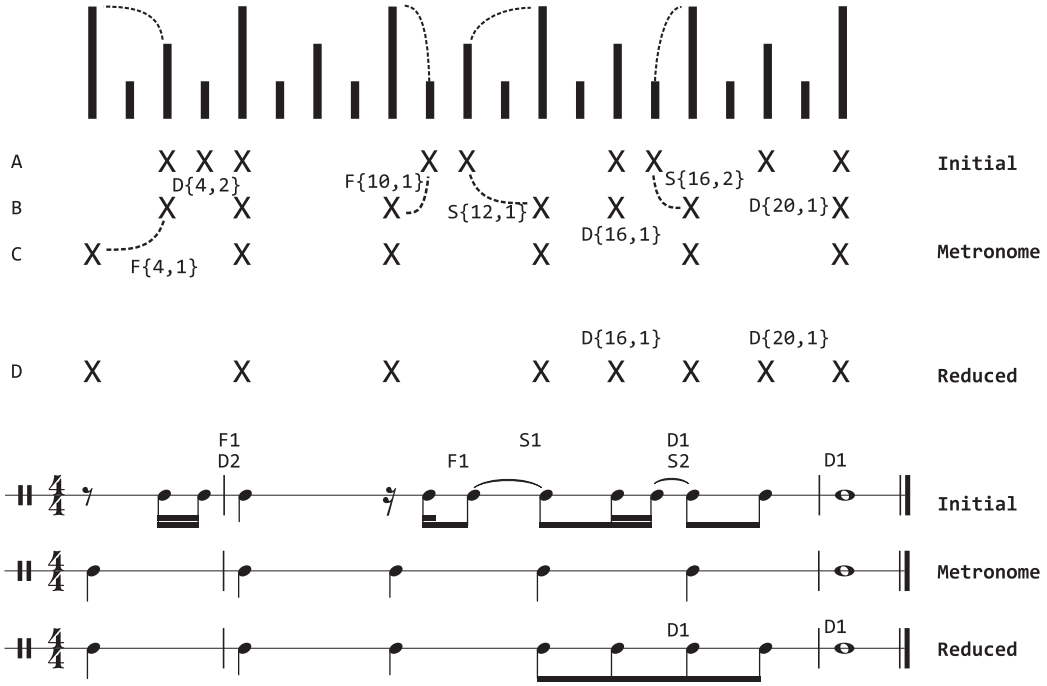


Figure 8. An example of a pattern of five beats, starting with an anacrusis, is characterized by the transformations: *D* {4, 2} *F* {10, 1} *S* {12, 1} *S* {16, 2} *D* {20, 1} *F* {4, 1} *D* {16,1}. Only the events (X) are shown in the binary representation; the dots (.) corresponding to the silent pulses are omitted for clarity. On the Initial and Reduced staves, a simplified representation of the vectors, which does not contain the pulse index but only the kind of transformation (*F*: figural, *S*: syncopation, *D*: density) and the metrical level difference (e.g. *F* {4, 1} - > *F*1), is shown above the corresponding metrical positions.

In Figure 8, a pattern labelled 'Reduced' is shown. This pattern is obtained by only generating the possible density transformations on the metronome without generating any figural or syncopations resulting in the events being aligned in their metronomic positions. The reduced pattern is an intermediate step between the initial pattern and the metronome, which preserves non-explicit rhythmic features (e.g. harmony or melodic intervals) while discarding the syncopations and pickups. A further reduction of the density transformations may remove essential non rhythmic features in many patterns. For instance, removing a chord from a weak metrical position could drastically change the harmonic structure of the phrase.

The characterization process can be followed for any pattern until the metronome corresponding to the given metrical template is reached. The order in which the pulses are scanned and the transformations are found and reversed does not affect the final outcome. The recursive nature of the process together with the reversibility constraints ensure that the given pattern can be generated only by the set of transformations found during the characterization.

However, when generating the initial pattern from the metronome, some transformations depend on others and must be performed in the order they were found. In Figure 8, the syncopation *S* {12, 1} must be performed before the figural *F* {10, 1}, which depends on the existence of the syncopating note on pulse 10. On the other hand, other transformations, such as the *D* {20, 1}, can be applied in any order without affecting the outcome. In general, transformations that share at least one event in their weak–strong pairs form compound transformations and their shifts must be performed in a certain order to ensure the reconstruction of the pattern. Another type of dependence between transformations that do not share a common event is found between *D* {16, 1} and *S* {12, 1}. Although each transformation does not depend on the existence of the other, when both are generated, they must be in the correct order; otherwise the density transformation would not be possible. Because of the reversibility constraints, the characterization algorithm can only reverse a transformation if no other transformation depends on it, thus ensuring that the result of the characterization process will always respect the order of the dependence of the transformations.

All the patterns that originate from the same metronome will lie on a tree-like lattice similar to the syncopation branches and trees presented by Sioros and Guedes (2014). In Figure 9, we present an example of such a 'rhythm' tree. The example illustrates in practice two important aspects of the model and how the reversibility constraints result in a unique characterization of the

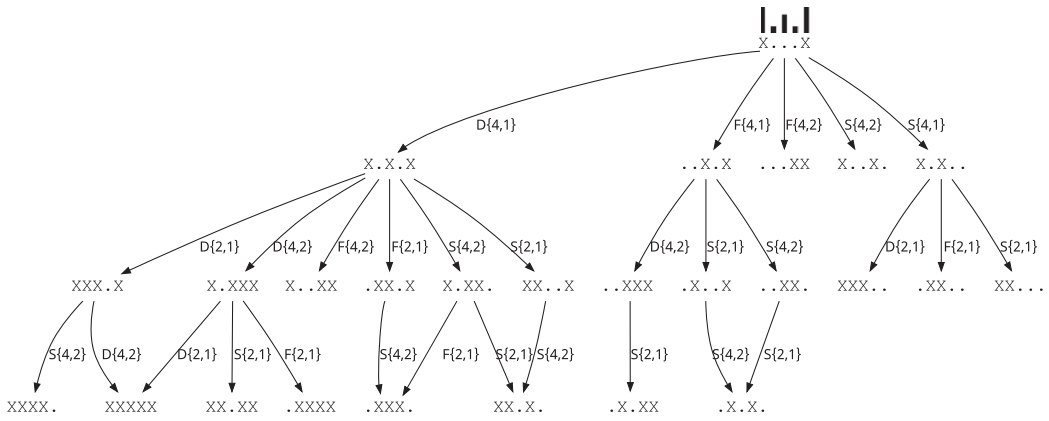


Figure 9. An example of a tree structure consisting of all the patterns generated by a metronome of a single beat duration corresponding to the metrical template shown with bold lines above the root of the tree.

patterns. First, all patterns that can be generated by a metronome will be found on the same tree. Second, certain patterns can be generated through more than one branch. Nevertheless, these branches always share the exact same transformations in a different order.

All the allowed transformations are represented by arrows connecting two patterns. Generating the transformations follows the direction of the arrows; characterizing a pattern follows the opposite direction. As some patterns can be the result of more than one independent transformation, which transformation is reversed first depends on which pulse we chose to characterize first. In any case, a complete characterization of a pattern follows the entire path until the metronome is reached and all transformations are reversed. These properties are general to the model and not specific to the example; they are common for trees constructed for metronomes with any metrical subdivisions (binary, ternary etc.) and for any concatenation of beat durations.

Sioros and Guedes (2014) presented an algorithm for de-syncopating any pattern and obtaining its non-syncopating counterpart called the *root*. Similarly, the characterization process presented here generates a branch of a tree connecting the pattern to a metronome. The rhythm tree resembles the syncopation tree of Sioros and Guedes with two important differences: (1) the root of the tree is not merely a non-syncopating rhythm but a metronome that articulates only beat positions of the given metre and (2) the branches hold every possible pattern with the same number of beats and not a small collection of patterns with same number of events and similar rhythmic figures.

3. Don't stop till you get enough

In this section, we demonstrate some of the strengths of the model by characterizing and discussing the main rhythmic pattern of the song 'Don't stop 'till you get enough' by Michael Jackson. The piece has several interesting properties which make it well suited for demonstrating the strengths of our approach. In Figure 10, the transformations of the original pattern to the metronome are shown, passing through the reduced pattern that contains only density transformations. For clarity, only the simplified transformation vectors are used in the figure, in which the single numerical component is the metrical level difference. The pulse index is implied by the positioning of the vector at the corresponding metrical position (e.g. *D*{12, 1} - > *D*1 above the third beat of the melody).

The two bars of the melody share several similarities and some important differences. Examining the melody, we can see three repetitions of a pattern of two sixteenth notes on weak metrical positions (marked with a circle on pulses 6, 22 and 30). The two first repetitions are identical in both their metrical positions and note durations. The third repetition differs only in the duration of the second note. The characterization of the transformations tells us that each repetition has a different character. In the first one, the notes are regarded as an anacrusis of the eighth note position leading to the following beat (transformation *F*1). The second repetition has the character of a metronome that speeds up as it runs on faster metrical levels as we approach the following beat. The two density transformations *D*1 and *D*2 articulate the two faster metrical levels but no other elements are introduced. The third repetition is regarded different as well, this time as a syncopation (*S*2) to the 'one' following on the next downbeat. It is notable that the third repetition is not an anacrusis even though the previous beat position (pulse 28) does not carry an event (in contrast, in the first repetition the same lack of event in the previous beat leads to the characteristic pickup feel). The above differences in the character of each repetition are encoded as the different transformations that generate each repetition.

Examining the short melodic figure (A-C#-F#) that appears twice, once in each bar, we can observe how figural transformations and syncopations have the characteristic

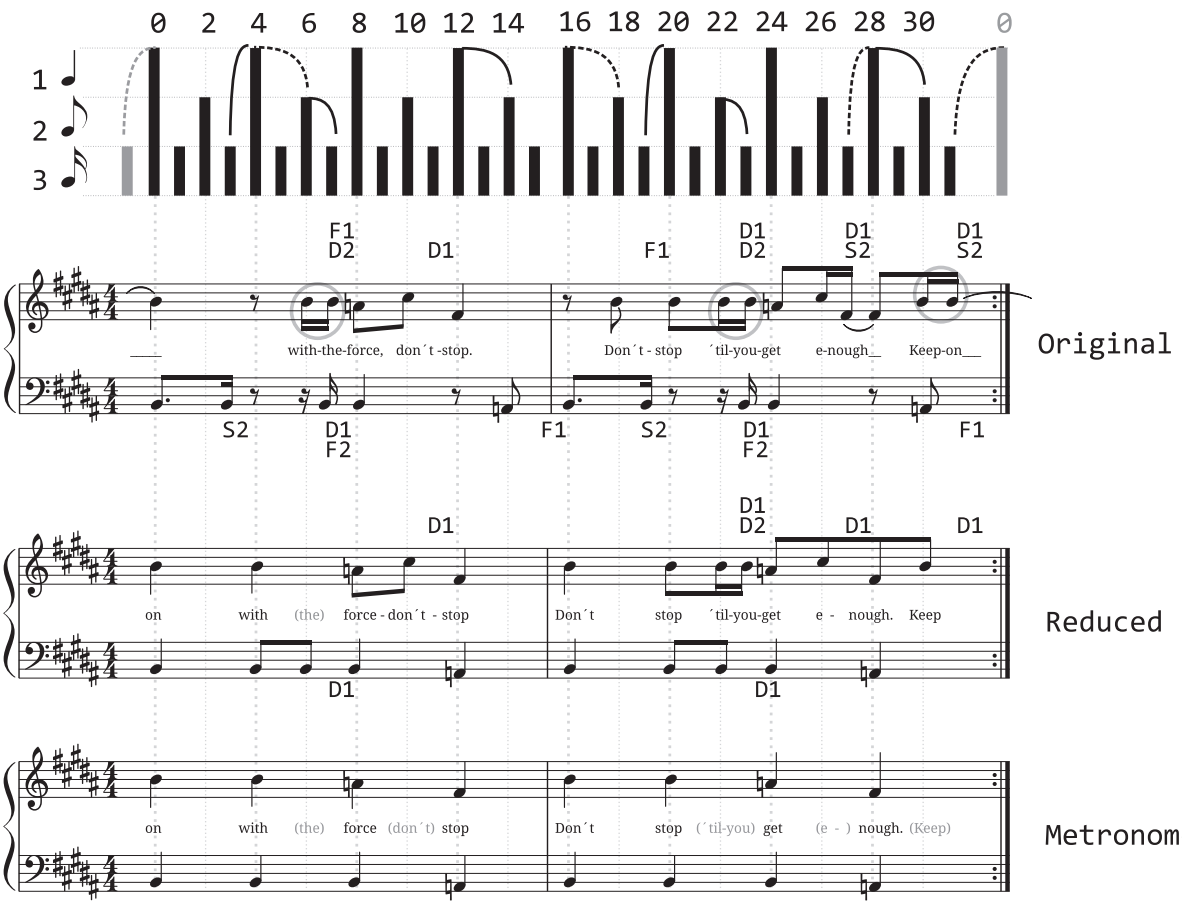


Figure 10. Analysis of the main rhythmic pattern (bass line and voice) of 'Don't stop till you get enough' by Michael Jackson. On the staves, a simplified representation of the vectors is shown where the pulse index is omitted as it is implied by the positioning of the vector at the corresponding metrical position (e.g. *F* {8, 2} - > *F*2 at the third beat of the bass).

effect of emphasizing the corresponding strong positions while density transformations have a less characteristic effect. The first time the melodic figure appears, emphasis is put on the second beat of the bar through the pickup rhythmic figure *F* {8, 1}. At the same time, the density transformations resemble a slowing down metronome that runs on slower and slower levels during the second half of the bar (*D*2 - > *D*1 - > No transformation). In the second bar, the syncopation on pulse 28 and the lack of a pickup in the previous beat (pulse 24) take the emphasis towards the third beat of the bar. In other words, replacing the figural transformation with a density transformation on the second beat and introducing a syncopation in the third beat, effectively shifts the stress from one to the other.

The bass line emphasizes each beat in a different way (except the last beat in each bar which is not transformed), giving no clear evidence on where the downbeat is located; a feature characteristic of the 'disco' feeling (Danielsen, 2012, pp. 156–157). The melody on the other hand, has a strongly emphasized downbeat more characteristic of funk (Danielsen, 2012, pp. 158–159). The syncopation *S*2 on the downbeat of the first bar is followed by the pickup *F*1 two beats later, in the middle of the bar, resulting in a clear and strongly felt metrical hierarchy. Beats are emphasized in a complementary way in the bass and melody. When there is a syncopation in one, there is a figural (or no transformation) in the other, but no syncopation coincides in both rhythmic layers, resulting in a very clear metrical feel.

The same principles could easily be applied to other pieces of music. Conclusions such as the above can be directly and systematically drawn through the transformation vectors without the need to examine the actual rhythmic patterns. The transformation vectors show the characteristic elements of the rhythm, on which beats and metrical positions emphasis and stress is put and how the rhythmic layers relate to each other throughout the duration of the music.

4. Discussion

In this paper, we have presented a model for the analysis and the characterization of rhythms that codifies patterns as transformations of a simple metronome that articulates only beat position. The transformations are categorized as (1) syncopation, (2) figural, which create anacruses and (3) density which insert events. The formalization of the transformations was based on the transformation vector, a concept previously introduced by Sioros and Guedes (2014). The characterization of a rhythm is done by reversing the transformations found in a pattern, through an algorithm that identifies them and outputs the corresponding vectors. The algorithm ensures that a single series of transformations describes a unique rhythmic pattern.

The model consists of transformations that take the form of shifts of events. While these shifts are not intended to reflect the composer's or performer's methods and intentions, they serve as a means to characterize and interpret a rhythmic pattern with regard to the metrical expectations of a listener. Each shift in the model introduces a specific musical phenomenon in the rhythm which contributes to its character. For instance, anticipating an event in a weak metrical position introduces a syncopation. In this way, the shifts are used as an effective technique to reach a meaningful reduction of the rhythm. At the same time, they preserve other properties and qualities of the events besides the timing, such as the pitch, timbre or lyrics.

The transformations were inspired by the syncopation model proposed by Temperley (1999). In his model, a pattern is represented by a music surface that is a transformation of an underlying deep structure. The two are connected by shifts of certain notes, from their metronomic positions in the deep structure to offbeat syncopating positions in the music surface. According to Temperley, transformations such as the syncopation shifts have a local character and a strong cognitive ground in comparison to other Schenkerian approaches to rhythm reduction (Komar, 1971; Schachter, 1980). Temperley draws evidence from music theory, harmonic analysis and rhythm perception to suggest that an unsyncopated deep structure and a syncopated musical surface correspond to actual cognitive representations of rhythm. In our model, the cognitive mechanism of weak–strong bonds described by Huron (2006, p. 295) lies behind such shifts.

On the same grounds, we speculate that the transformations, the metronome and the reduced pattern provide meaningful rhythm representations that correspond to listeners' interpretations of rhythms. Different listeners form different metrical expectations in response to a rhythm that affect and determine their experience of music. In our model, the listener's metrical expectations are modelled in metrical templates that constraint the transformations and determine the characterization of a rhythm. Thus, several characterizations for the same pattern are possible that reflect different listeners' expectations. The automatic construction of templates of Section 2 is limited to common time signatures, nevertheless any well-formed metre, including non-isochronous metres (London, 2012), can be expressed in a metrical template. However, a systematic evaluation of the model with respect to rhythm perception is outside the scope of this article and is left for a future study.

Our model also shares similarities with the generative models developed by Temperley (2009, 2010) and his metrical anchoring principle. Anchoring characterizes events in weak metrical positions depending on the existence of events in their surrounding stronger pulses as: (1) both-anchored when events exist in both the previous and the following stronger pulses, (2) post-anchored when only the following pulse contains an event, (3) pre-anchored when only the preceding pulse contains an event and (4) unanchored when neither the receding nor the following stronger pulses contain events. Accordingly, a syncopation transformation generates a pre- or an un-anchored event, a figural transformation a post-anchored event and a density transformation a both-anchored event.

However, the two models are not equivalent as they focus on different aspects of rhythm. Temperley (2010, p. 372) views pre- and unanchored notes as modelling the syncopation in patterns, but syncopation itself is as an overall quality of the rhythm related to the likelihood of a rhythm to appear in a certain metrical context. A pattern is felt as more syncopated than another due to a higher number of unanchored notes, but those notes do not necessarily correspond to instances of syncopations. Similarly, post- and both-anchored notes contribute to the overall metrical feel but are not viewed as instances of a rhythmic phenomenon. In contrast, our transformations model and generate specific phenomena that have a local character. Syncopations and anacruses are felt at particular moments and have short durations. The overall metrical and syncopation feel could be attributed to their combinations in a rhythm, but such a mapping would depend on cultural factors or the musical context and style.

For instance, the sixteenth-note found on pulse 30 of 'Don't stop 'till you get enough' is unanchored as it is not surrounded by events in the stronger metrical positions. A direct equivalence between anchoring and syncopation would lead to the wrong conclusion that the note on pulse 30 is syncopated. In our model, this is attributed to the syncopations on the surrounding beats. The note simply *happens* to occur between two syncopating sixteenth notes at pulses 27 and 31 which are pre-anchored.

The set of transformations that correspond to a rhythm carry all its characteristic features with respect to the given beat. On the one hand, the combination of syncopations and pickups describe the timing of the events in relation to certain metrical expectations. On the other hand, the density transformations articulate fast metrical levels as a metronome that varies its speed running at different metrical levels. The reduced pattern represents such a variable metronome. The specific combination of the three elements makes each pattern unique. For instance, in the 'Don't stop 'till you get enough' example presented in Section 3, the two different bars of the melody originate from the exact same metronome. Their differences are encoded into the different sets of transformations that generate them. Motifs with similar inter-onset intervals on the musical surface, such as the repeated sixteenth notes, become distinct as their individual feel and character is brought to the foreground through the characterization process.

The encoding in transformation vectors gives the model and the representation of transformations great flexibility. On the staves of Figures 8 and 10, the transformation vectors are simplified by omitting the pulse index and showing only the metrical level difference. The pulse is implied by their placement above the corresponding metrical position. This simplification of the vector reveals an important aspect of its nature. After the characterization of a pattern is complete, the transformation vectors can be separated from the metrical template. Their metrical position can be referenced using appropriate references for each music material such as the beat number together with phrase boundaries or simply placing them in a time line as in Figures 8 and 10.

A similar encoding based on syncopation transformations alone was initially developed by Sioros and Guedes (2014). Our model is based on the same concept of transformation, extending it to include the density and figural transforms, and thus resulting in a complete model for binary or quantized musical rhythms. The introduction of the two transformations has a strong impact on the syncopation transformations as well which is evident in the example in Figure 7. This earlier syncopation model allowed for consecutive syncopations to mask or hide one another, which posed no problem considering the focus on generative applications. However, in our complete model, the focus is shifted towards a more meaningful and accurate characterization of rhythms, and such 'hidden' syncopations are avoided using a combination of figural and syncopation transformations.

While this paper introduces the model and its general aspects as a method for the analysis of musical rhythms and not any of its specific applications, we now present some of potential uses and future developments. The automatic and systematic encoding of each rhythmic pattern into a unique set of transformations can be of great value in musicological analysis. A systematic and automatic comparison and classification of music excerpts as well as measures of similarity or distance between patterns can be based on the common transformation vectors the excerpts share. As opposed to the more abstract mathematical properties used in the measures compared by Toussaint (2004), the transformation vectors describe music elements such as syncopations and pickups, rather than distance and statistical properties of binary rhythmic representations. Their comparison should depend on the intended applications and appropriate weights should be used according to the music material in question. On the other hand, the vectors and tree structures are not designed to describe an optimal transformational path between two patterns in the fashion of the edit distance measure proposed by Post and Toussaint (2011). As we discuss in Section 2.3, the reversibility constraints impose a certain order for the transformations which could force the path between rhythms to pass through unnecessary intermediary patterns, but with the advantage of providing a unique characterization for each pattern.

Prototypical transformations characteristic of a style or genre can be modelled and used in the classification of rhythms according to whether they share the same branch. Even the relation between different styles can be explored by examining the transformation path between style prototypes.

The rhythmic variations that are generated as a by-product of the characterization of a pattern can assist the automatic analysis and comparison of other properties of the events. For instance, in the 'Don't stop till you get enough' example, we can directly observe in the metronome and reduced pattern that the harmony and melodic structure of the two bars is identical. Their differences are the result of a different rhythmic interpretation of the melody. Such analyses and comparisons can be facilitated and automated for any multi-layered rhythm. The notes of the various rhythmic layers are aligned in the generated metronomes and reduced patterns, so that their relations are simplified. A systematic analysis of these simplified patterns instead of the complex and often idiosyncratic original rhythms is easier and more effective.

In conclusion, the model presented here effectively combines powerful features and has a great potential in music analysis. It codifies rhythms in a systematic and flexible way, while at the same time provides with a musically and perceptually meaningful and detailed characterization, and the generative nature of the model has added value beyond the scope of rhythm analysis.

Acknowledgements

We would like to express our appreciation to the NYUAD Institute for organizing the three editions of the 'Cross-disciplinary and multicultural approaches to music rhythm' workshop and to Roger Dannenberg, Bill Sethares and all the participants who provided insights and expertise that greatly improved this research.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by North Portugal Regional Operational Programme (NORTE 2020) Project TEC4Growth-Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact [grant number NORTE-01-0145-FEDER-000020], PORTUGAL 2020 Partnership Agreement, the European Regional Development Fund (ERDF); Matthew E.P. Davies is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P. [grant number IF/01566/2015].

ORCID

Matthew E. P. Davies http://orcid.org/0000-0002-1315-3992
Carlos Guedes http://orcid.org/0000-0002-1898-2183

References

Barlow, C. (1987). Corrections for Clarence Barlow' s article: Two essays on theory. *Computer Music Journal, 11*(4), 10.

Barlow, C., & Lohner, H. (1987). Two essays on theory. *Computer Music Journal, 11*(1), 44–60.

Clarke, E. F. (1987a). Categorical rhythm perception: An ecological perspective. In A. Gabrielsson (Ed.), *Action and Perception in Rhythm and Music* (pp. 19–33). Stockholm: Royal Swedish Academy of Music.

Clarke, E.F. (1987b). Levels of structure in the organization of musical time. *Contemporary Music Review, 2*(1), 211–238. doi:10.1080/07494468708567059

Danielsen, A. (2012). The sound of crossover: Micro-rhythm and sonic pleasure in Michael Jackson's "Don't stop "Til you get enough"". *Popular Music and Society, 35*(2), 151–168. doi:10.1080/03007766.2011.616298

Fraisse, P. (1982). Rhythm and Tempo. In D. Deutsch (Ed.), *The Psychology of Music* (1st ed.). (pp. 149–180). New York, NY: Academic Press.

Honing, H. (2012). Structure and Interpretation of Rhythm in Music. In D. Deutsch (Ed.), *The Psychology of Music* (3rd ed.). (pp. 367–404). Academic Press, Elsevier.

Huron, D. (2006). *Sweet anticipation: Music and the psychology of expectation*. Cambridge, MA: The MIT Press.

Huron, D., & Ommen, A. (2006). An empirical study of syncopation in American popular music, 1890–1939. *Music Theory Spectrum, 28*(2), 211–231. doi:10.1525/mts.2006.28.2.211

Jones, M. R. (2008). Musical time. In S. Hallam, I. Cross, & M. Thaut (Eds.), *The oxford handbook of music psychology* (pp. 81–92). New York, NY: Oxford University Press. https://doi.org/10.1093/oxfordhb/9780199298457.013.0008

Jones, M.R., Moynihan, H., MacKenzie, N., & Puente, J. (2002). Temporal aspects of stimulus-driven attending in dynamic arrays. *Psychological Science, 13*(4), 313–319. doi:10.1111/j.0956-7976.2002.00458.x

Komar, A.J. (1971). *Theory of suspension*. Princeton NJ: Princeton University Press.

Lerdahl, F., & Jackendoff, R. (1981). On the theory of grouping and meter. *The Musical Quarterly, 67*(4), 479–506.

Lerdahl, F., & Jackendoff, R. (1983). *A generative theory of tonal music*. Cambridge, MA: The MIT Press.

Lewin, D. (2007). *Generalized Musical Intervals and Transformations*. New York, NY: Oxford University Press. doi:10.1093/acprof:oso/9780195317138.001.0001

London, J. (2012). *Hearing in Time* (2nd ed.). New York, NY: Oxford University Press.

Longuet-Higgins, H. C., & Lee, C. S. (1984). The rhythmic interpretation of monophonic music. *Music Perception: An Interdisciplinary Journal, 1*(4), 424–441. doi:10.2307/40285271

Louboutin, C., & Meredith, D. (2016). Using general-purpose compression algorithms for music analysis. *Journal of New Music Research, 45*(1), 1–16. doi:10.1080/09298215.2015.1133656

McAuley, J.D. (2010). Tempo and Rhythm. In M. Riess Jones, R.R. Fay, & A. N. Popper (Eds.), *Music Perception* (pp. 165–199). New York, NY: Springer. https://doi.org/10.1007/978-1-4419-6114-3

Meredith, D. (2014). Compression-based geometric pattern discovery in music. *4th International Workshop on Cognitive Information Processing – Proceedings of CIP*. Retrieved from https://doi.org/10.1109/CIP.2014.6844503

Paiement, J.-F., Grandvalet, Y., Bengio, S., & Eck, D. (2007). A generative model for rhythms. In *Music, brain, and cognition workshop, NIPS 2007*. IDIAP.

Palmer, C., & Krumhansl, C.L. (1990). Mental representations for musical meter. *Journal of Experimental Psychology. Human Perception and Performance, 16*(4), 728–741.

Parncutt, R. (1987). The perception of pulse in musical rhythm. In A. Gabrielsson (Ed.), *Action and Perception in Rhythm and Music* (pp. 127–138). Stockholm: Royal Swedish Academy of Music.

Parncutt, R. (1994). A perceptual model of pulse salience and metrical accent in musical rhythms. *Music Perception: An Interdisciplinary Journal, 11*(4), 409–464.

Post, O., & Toussaint, G. (2011). The edit distance as a measure of perceived rhythmic similarity. *Empirical Musicology Review, 6*(3), 164–179.

Povel, D.-J., & Okkerman, H. (1981). Accents in equitone sequences. *Perception & Psychophysics, 30*(6), 565–572. doi:10.3758/BF03202011

Repp, B.H. (2006). Rate Limits of Sensorimotor Synchronization. *Advances in cognitive psychology, 2*(2), 163–181. doi:10.2478/v10053-008-0053-9

Schachter, C. (1980). Rhythm and linear analysis: Durational reduction. In F. Salzer (Ed.), *The Music Forum 4*. New York, NY.

Sioros, G, & Guedes, C. (2014). Syncopation as Transformation. In M. Aramaki, O. Derrien, R. Kronland-Martinet, & S. Ystad (Eds.), *Sound, Music, and Motion* (Vol. 8905, pp. 635–658). Cham: Springer International. https://doi.org/10.1007/978-3-319-12976-1

Temperley, D. (1999). Syncopation in rock: A perceptual perspective. *Popular Music, 18*(1), 19–40. doi:10.1017/S0261143000008710

Temperley, D. (2009). A Unified Probabilistic Model for Polyphonic Music Analysis. *Journal of New Music Research, 38*(1), 3–18. doi:10.1080/09298210902928495

Temperley, D. (2010). Modeling Common-Practice Rhythm. *Music Perception, 27*(5), 355–376.

Toussaint, G. (2002). A Mathematical Analysis of African, Brazilian and Cuban Clave Rhythms. In R. Sarhangi (Ed.), *Bridges: Mathematical Connections in Art, Music, and Science* (pp. 157–168). Towson, MD: Bridges Conference.

Toussaint, G. T. (2004). A comparison of rhythmic similarity measures. In *Proceedings international conference on music information retrieval (ISMIR 2004)* (pp. 242–245).

Toussaint, G. T. (2013). *The geometry of musical rhythm: What makes a 'good' rhythm good?* Chapman and Hall/CRC.

Yeston, M. (1976). *The stratification of musical rhythm*. New Haven, CT: Yale University Press.

2019 27th European Signal Processing Conference (EUSIPCO)

Temporal convolutional networks for musical audio beat tracking

Matthew E. P. Davies
INESC TEC
Porto, Portugal
matthew.davies@inesctec.pt

Sebastian Böck
Austrian Research Institute for Artificial Intelligence (OFAI)
Vienna, Austria
sebastian.boeck@ofai.at

***Abstract*—We propose the use of Temporal Convolutional Networks for audio-based beat tracking. By contrasting our convolutional approach with the current state-of-the-art recurrent approach using Bidirectional Long Short-Term Memory, we demonstrate three highly promising attributes of TCNs for music analysis, namely: i) they achieve state-of-the-art performance on a wide range of existing beat tracking datasets, ii) they are well suited to parallelisation and thus can be trained efficiently even on very large training data; and iii) they require a small number of weights.**

***Index Terms*—Beat Tracking, Music Signal Processing, Convolutional Neural Networks**

I. Introduction

The task of musical audio beat tracking has been well-established over the last twenty-five years [1]. While the goal of estimating a sequence of quasi-periodic time instants to reflect how a human listener would synchronise their taps to the beat of the music has remained largely unchanged, there has been a shift away from purely signal processing-based approaches to those incorporating machine learning, and most recently deep learning. One means of understanding this change is to consider how early approaches to beat tracking relied on the use of onset strength functions as the primary input representation. Given such an input containing peaks at onset locations, the aim was to identify and track a latent periodicity and subsequently (or simultaneously) identify the subset of these peaks most likely to correspond to the beat of the music. In this sense, the aim was to recover a hidden sequence of beats from an observed representation related to musical onsets.

In an effort to filter out peaks that were unlikely to correspond to beats, Davies *et al.* [2], derived a so-called beat emphasis function as the linear combination of sub-band onset strength functions weighted by their respective beat strength. While effective in enhancing periodic peaks in the onset strength functions it yielded only a moderate improvement over existing state of the art approaches (*e.g.*, [3]). A radically different approach was proposed by Böck and Schedl [4] who reformulated the beat tracking task using Recurrent Neural Networks (RNNs), specifically, a Bidirectional Long Short-Term Memory (BLSTM) model, to output a beat activation function (with peaks only at beat locations) given a log magnitude spectrogram input representation and a training dataset of manually annotated beat locations. In this way, a detected sequence of beat locations could be obtained simply by peak-picking the beat activation function.

Limitations in both the amount of training data and the variable quality of the annotations led to more sophisticated, and ultimately more successful, approaches for obtaining beat times from *imperfect* beat activation functions. These included the selection between multiple trained models adapted to different types of musical content and the use of a dynamic Bayesian network (DBN) for decoding the beat activation function [5], with further gains possible by the combined modelling of beat and downbeat information [6], a step that echoes the probabilistic approach of Klapuri *et al.* [3] which simultaneously estimated beat, downbeat, and tatum levels.

In spite of the inclusion of more sophisticated post-processing applied to the beat activation function and the considerable improvements obtained from access to more training data, the core BLSTM approach [4] has remained essentially unchanged, perhaps due to the inherent modelling power of recurrent models for sequential data. Nevertheless, recurrent models are very hard to train and possess certain limitations, including: the vanishing gradient problem, difficulty in interpreting the different internal layers of the model, and a learning approach which doesn't lend itself to efficient parallelisation using GPUs.

Convolutional Neural Networks (CNNs) on the other hand, and particularly for image processing tasks, are amenable to exposing the representations held in different layers [7]. Furthermore, CNNs are highly parallelisable and can thus be trained very efficiently on GPUs. For the task of beat tracking they have only been deployed to predict local features (*e.g.*, [8]) or model beat sequences with (large) filters which extend over a context of several hundred frames [9]. Recently, convolutional recurrent neural networks (CRNNs) have emerged in an attempt to leverage the modelling power of both CNNs and RNNs, where recurrent layers are attached to the output of convolutional models [10], [11].

In this paper, we explore the ability of Temporal Convolutional Networks (TCNs) [12] for a sequential learning task,

Matthew E.P. Davies is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the project IF/01566/2015. Sebastian Böck is supported by the Austrian Promotion Agency (FFG) under the "BASIS, Basisprogramm" umbrella program.

978-9-0827-9703-9/19/$31.00

namely musical audio beat tracking. Having first appeared in the well-known *WaveNet* model [13], TCNs perform dilated convolutions (*i.e.*, convolutions across sub-sampled input representations) for learning sequential/temporal structure. In this way, they retain the parallelisation property of standard CNNs, and have been shown to outperform recurrent approaches on a range of sequential learning problems [12]. While the *WaveNet* approach used the raw audio waveform as input, we reformulate the current state-of-the-art approach for beat tracking [6], by substituting the BLSTM with a TCN and applying it to an input representation derived from a log magnitude spectrogram. In doing so, we demonstrate the TCN approach is able to perform on par with the state of the art on existing annotated beat tracking datasets, that it can be trained far more efficiently from a computational perspective, and it also benefits from a very small number of weights $(21,809)$ which is in stark contrast to many existing deep learning approaches, which can have millions of trainable parameters, and roughly a third of the BLSTM method [4].

The remainder of this paper is structured as follows. In Section II we describe our TCN approach for beat tracking. This is followed in Section III by an objective evaluation against the current state of the art. In Section IV we discuss the impact of our approach and propose areas for future work.

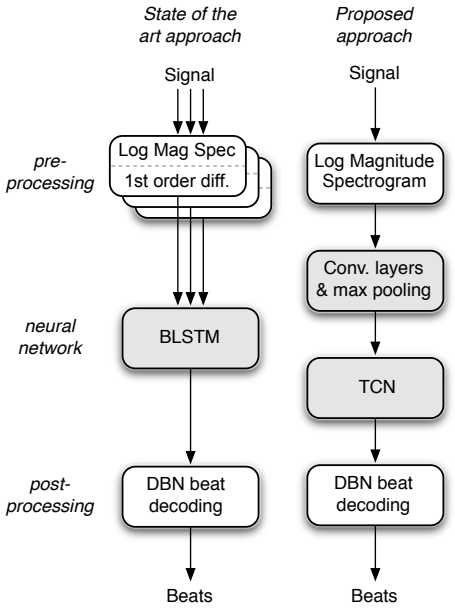


Fig. 1. Comparison between existing state of the art (left) with our proposed approach (right). The neural network blocks are shaded light grey.

II. Approach

A. Overview of existing state of the art

We base our approach around the model first presented by Böck and Schedl [4], and later extended in [5], [6] whose main processing pipeline is shown in the left hand side of Fig. 1. Given a mono audio input signal, sampled at $44.1\,\text{kHz}$, the input representation is derived from a set of log magnitude spectrograms which are grouped to have approximately logarithmic frequency spacing between adjacent bins. Three such spectrograms are calculated at a fixed hop size of $10\,\text{ms}$ with increasing window sizes of $23.2\,\text{ms}$, $46.4\,\text{ms}$ and $92.9\,\text{ms}$. From each, the per-bin first-order difference spectorgram is calculated, where only the positive differences are retained to capture the energy rise in individual frequency bands. All of these spectrogram representations are vertically stacked, and this multi-resolution input representation is then passed to a three layer BLSTM, with each layer having 25 recurrent units. In [4], the final beat locations were obtained by peak picking the beat activation function, however in [5], this processing step was replaced by a DBN approximated via a hidden Markov model (HMM). For more details see [4], [5].

B. Proposed reformulation using TCNs

An overview of our proposed approach is shown in the flow chart on the right of Fig. 1. By comparing the two processing pipelines, that of the existing state of the art (left), and our proposed method (right), we can observe: i) a simplification in the input representation, which replaces six spectrogram type inputs with just one and thus greatly reduces the input dimensionality; and ii) the inclusion of a set of convolution and max pooling layers prior to the main sequence learning model, the TCN, which replaces the BLSTM network. Our goal here is to minimise the need for explicit design choices in the front-end to our beat tracker, and thus place greater emphasis on what can be learned from a "simpler" input representation.

1) Input and Target Representations: As the initial input representation we use a single log magnitude spectrogram with a hop size of $10\,\text{ms}$ and a window size of $46.4\,\text{ms}$ (2048 samples). A logarithmic grouping of frequency bins with 12 bands per octave provides an input representation with a total of 81 frequency bands from $30\,\text{Hz}$ up to $17\,\text{kHz}$, as summarised in the *Signal Conditioning* section of Table I.

In the context of beat tracking, we are seeking to predict a beat activation function from this input representation which exhibits peaks at likely beat locations and can be used to recover an output sequence of beat times. To this end, we treat the beat tracking problem as a binary classification task, where annotated beat locations are first quantised to the temporal resolution of the input representation, and then represented as training targets. The goal is then to predict the likelihood of a beat occurring at any given time frame of the log magnitude spectrogram. Following the strategy of [14] for onset detection, we *widen* the temporal activation region around the annotations to include two adjacent temporal frames on either side of each quantised beat location and weight them with a value of 0.5 during training.

2) Convolutional Block: While the log magnitude spectrogram could be passed directly to the TCN, we first seek to learn some compact intermediate representation. To this end, we employ three convolutional layers: the first two layers with 16 filters of size 3×3 with subsequent max pooling over 3 bins in the frequency direction; and a third with 16 filters of

TABLE I
OVERVIEW OF SIGNAL PROCESSING AND LEARNING PARAMETERS

Signal Conditioning	
Audio sample rate	44.1 kHz
Window shape	*Hann*
Window & FFT size	2048 *samples*
Hop size	10 ms
Filterbank freq. range	30 . . . 17000 Hz
Sub-bands per octave	12
Total number of bands	81
Conv. Block	
Number of filters	16, 16, 16
Filter size	3×3, 3×3, 1×8
Max. pooling size	1×3, 1×3, —
Dropout rate	0.1
Activation function	*ELU*
TCN	
Number of stacks	1
Dilations	$2^{0,\ldots,10}$
Number of filters	16
Filter size	5
Spatial dropout rate	0.1
Activation function	*ELU*
Training	
Optimizer	*Adam*
Learning rate	0.001
Batch size	1
Output activation function	*sigmoid*
Loss function	*binary cross-entropy*

size 1×8 without pooling. In this way, small (overlapping) spectrogram snippets with a context of 5 frames get reduced to a single frame and 16 features. The exponential linear unit (*ELU*) [15] is used as activation function in the convolutional layers, and a dropout [16] rate of 0.1 applied afterwards. All parameters are summarised in the *Conv. Block* section of Table I. By learning these filters within the network we can derive an intermediate representation which is better adapted to the input data and much smaller than the hard-coded choice of the bin-wise temporal difference.

3) Temporal Convolution Network: The principal means by which the TCN is able to capture sequential structure is by learning filters via *dilated* convolutions. In our case, the input to the TCN is a 16-dimensional feature vector derived from the magnitude spectrogram by the convolutional block, which retains the same temporal resolution. The learning target for the TCN is to predict the beat locations from annotated training data as described in Section II-B1. By working on a highly sub-sampled feature representation compared to the raw audio, we can obtain a large temporal receptive field with far fewer layers and weights than the raw audio domain equivalent.

The TCN, as presented in [13], is highly parameterisable, with the principal degrees of freedom being: the number of TCN filters, the kernel size (*i.e.*, shape of the filters), the number of layers, their dilation rates, and the number of times the model can be stacked. While the number of filters and their shape are quite standard properties of CNNs, the number of dilations, their rate, and the number of stacks of the model, are what contribute to the width of the receptive field of the model. The TCN illustration shown in Fig. 2 contains four layers with an exponentially increasing dilation rate and demonstrates how the output depends on relations with time points which are potentially quite distant. In contrast to RNN approaches, they are not sequentially connected. Indeed, it is this lack of RNN-like long-term sequential connections which contribute to the highly parallelisable structure of the TCN, and thus drastically increase the computational efficiency when training on GPUs.

Two important distinctions between our TCN and the original formulation [13] are that we replace all activation functions within the TCN with *ELUs* [15], and modify it to operate non-causally, rather than in a purely causal way as defined in [12]. In practice, the latter modification means that for any given temporal frame of the input, the (dilated) convolutions extend both forwards and backwards in time. For purely causal operation the dilated convolutions are only performed using past data up to the current temporal frame with no access to future information in the signal. In the context of real-time beat tracking, such causal processing would be essential (as well as the need to adapt many other components of the beat tracking system), but for this paper where all other processing steps are performed offline, we allow full access to the input signal, and seek to benefit from the additional temporal context provided by the non-causal convolutions.

Concerning the specific parameterisation of our TCN, we have attempted to strike a balance between high beat tracking accuracy and the simplicity of the model (*i.e.*, minimising the number of weights to learn). To this end, we propose the parameters shown in the *TCN* section of Table I, and leave a more thorough optimisation of the parameters as a topic for future work. We train the model with the *Adam* optimiser [17], a batch size of 1 and a learn rate of $1e^{-3}$. We reduce the learn rate by a factor of 5 if the loss on the disjoint validation set reaches a plateau and stop training if no improvement in the validation loss is observed for 50 epochs.

4) Obtaining beat predictions: The output of the TCN is a one dimensional beat activation function, intended to exhibit peaks at likely beat locations and be close to zero at all other points in time. For musical examples where the beat activation function approximates this idealised structure, the output can be obtained by a simple peak-picking process. In practice, the beat activation function often has peaks at non-beat locations, or fails to produce peaks at annotated locations, and thus peak-picking alone is typically insufficient to provide a plausible beat tracking output. To this end, [5] proposed using a DBN to decode the beat activation function which yields better global alignment of beats. In this work we use the enhanced and more efficient state space proposed by Krebs *et al.* [18]. Given the beat activation function produced by our TCN has the same temporal resolution and target structure as in [5], we directly reuse this existing DBN together with the default parameters given in [18]: a tempo range of 55–215 beats per minute, and the transition-λ, which aims to control the ability of the model to react to tempo changes, at a value of 100.

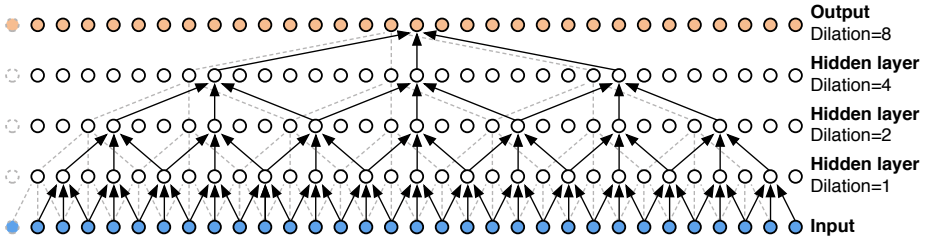


Fig. 2. Overview of the TCN structure (adapted from the original version [13]) to demonstrate non-causal operation. The grey dashed lines show the network connections shifted back one time step.

III. Experiments and Results

To determine the performance of our proposed TCN beat tracking system, we follow the *de facto* objective evaluation methodology by measuring beat tracking accuracy on annotated datasets and comparing it against state-of-the-art reference algorithms [19]. To permit the use of existing annotated datasets, both to train our model and measure its performance, we use 8-fold cross validation. We ensure the separation between testing and training data in each iteration of the cross validation, by using six folds for training, one for validation, and the remaining fold for testing, and rotate the folds eight times until each fold has uniquely been used for testing. The datasets used for cross validation are shown in the upper part of Table II. To provide some insight into the performance on totally unseen data, the *GTZAN* dataset [20] with the beat annotations from [21] is included as test data, but excluded from training, both for our TCN approach and the two reference state-of-the-art methods [5], [6].

TABLE II
Overview of the datasets used for training and evaluation.

Dataset	# files	length
Ballroom [22], [23] [1]	685	5 h 57 m
Beatles [19]	180	8 h 09 m
Hainsworth [24]	222	3 h 19 m
Simac [25]	595	3 h 18 m
SMC [26]	217	2 h 25 m
GTZAN [20], [21]	999	8 h 20 m

In Table III we list the beat tracking results using the widely adopted set of F-measure, continuity-based evaluation scores (CMLc, CMLt, AMLc, AMLt), and Information Gain (D), with the latter measured in bits. For further details, see [19]. Note that the results of [5] differ from the original publication since the DBN used for beat inference was updated to be the same as the one used in this paper which yields better performance and is computationally more efficient (cf. Section II-B4, [18]). Results on *GTZAN* were computed by averaging the predictions of all models trained with cross validation (*i.e.*, model bagging) before inferring the final beat locations with the DBN.

[1]We removed the 13 duplicates identified by Bob Sturm: http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html

TABLE III
Overview of beat tracking performance.

	F-measure	CMLc	CMLt	AMLc	AMLt	D
Ballroom						
TCN	0.933	0.864	0.881	0.909	0.929	3.456
BLSTM [5]	0.917	0.832	0.849	0.905	0.926	**3.539**
BLSTM [6]	**0.938**	**0.872**	**0.892**	**0.932**	**0.953**	3.397
Hainsworth						
TCN	0.874	0.755	0.795	**0.882**	**0.930**	**3.518**
BLSTM [5]	**0.884**	**0.769**	**0.808**	0.873	0.916	3.507
BLSTM [6]	0.871	0.732	0.784	0.849	0.910	3.395
SMC						
TCN	**0.543**	**0.315**	**0.432**	**0.462**	**0.632**	**1.574**
BLSTM [5]	0.529	0.296	0.428	0.383	0.567	1.460
BLSTM [6]	0.516	0.307	0.406	0.429	0.575	1.514
GTZAN						
TCN	0.843	0.695	0.715	0.889	0.914	**3.096**
BLSTM [5]	**0.864**	**0.750**	**0.768**	**0.901**	**0.927**	3.071
BLSTM [6]	0.856	0.716	0.744	0.876	0.919	3.019

Inspection of Table III demonstrates that across datasets and evaluation methods, the performance of our proposed approach is highly comparable to the existing state of the art, with this pattern holding both for those datasets included in the cross validation and the withheld *GTZAN* dataset. For the *Ballroom*, *Hainsworth*, and *GTZAN* datasets performance is on a very high level, irrespective of evaluation method. Conversely, the *SMC* dataset reveals a significant and expected drop in performance across all methods due to the large proportion of highly challenging musical excerpts. However, performance is highest for our proposed method.

Perhaps what is most noteworthy about our TCN approach is that it can maintain competitive performance but with two distinct computational advantages over the state of the art. Putting aside the approach in [6] which also estimates downbeats and is thus far more complex, our TCN approach uses fewer than 35% of the weights of the BLSTM [5] ($21,809$ vs. $67,301$). Furthermore, the learning of the TCN weights occurred at a rate of approximately 2 seconds/hour of audio on a recent GPU, compared to 2 minutes/hour of audio for the BLSTM, offering a 60x speed-up in training time. Put in context, this implies that our TCN can encode knowledge about the beat structure in music in a more compact representation than the BLSTM, and it can learn this information at a considerably

faster rate. This holds significant promise for learning on very large annotated datasets in practical computation time, *i.e.* in the order of hours rather than weeks, and therefore facilitating multiple training runs with different hyperparameter settings. One limitation of our proposed approach is that the inference is moderately slower than the BLSTM [5], but still much faster than real-time, processing 1 minute of audio in roughly 4–5 s on a recent laptop using only the CPU.

IV. Discussion and Conclusions

We have proposed a new approach for musical audio beat tracking using Temporal Convolutional Networks. Inspired by the well-known *WaveNet* generative model for raw audio signals [13], we re-purpose it to perform dilated convolutions along the temporal dimension of a jointly learned 16 dimensional feature vector in order to predict the locations of musical beats. Since the temporal resolution of the time-frequency representation is drastically lower than that of raw audio signals, this leads to a substantial decrease in the number of weights, which can be trained extremely efficiently, and is consequently less prone to overfitting.

In comparison with state-of-the-art recurrent beat trackers, we demonstrate that our TCN approach can achieve equivalent performance across a diverse range of annotated musical material, and improved performance on the dataset considered the most challenging for beat tracking. Furthermore, this high performance is embedded in a model which uses proportionally far fewer dimensions in its input representation by leveraging convolutional layers to implicitly learn a compact feature representation which is able to model the (local) temporal structure without explicitly encoding this information in the input representation, as done by existing recurrent approaches.

These promising initial results achieved with the TCN suggest significant potential for future work, including: learning the beats directly from the audio signal itself; simultaneously modelling beat and downbeats; developing a real-time approach using a causal TCN; and exploring TCNs for other time-based music analysis tasks such as chord recognition, note transcription, and structural segmentation.

References

[1] M. Goto and Y. Muraoka, "A beat tracking system for acoustic signals of music," in *Proc. of the 2nd ACM Intl. Conf. on Multimedia*, 1994, pp. 365–372.

[2] M. E. P. Davies, M. D. Plumbley, and D. Eck, "Towards a musical beat emphasis function," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2009, pp. 61–64.

[3] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions Speech and Audio Processing*, vol. 14, no. 1, pp. 342–355, 2006.

[4] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proc. of the 14th Intl. Conf. on Digital Audio Effects*, 2011, pp. 135–139.

[5] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles," in *Proc. of the 15th Intl. Society for Music Information Retrieval Conf.*, 2014, pp. 603–608.

[6] ——, "Joint beat and downbeat tracking with recurrent neural networks," in *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.*, 2016, pp. 255–261.

[7] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. of the European Conf. on computer vision*. Springer, 2014, pp. 818–833.

[8] S. Durand, J. P. Bello, B. David, and G. Richard, "Feature Adapted Convolutional Neural Networks for Downbeat Tracking," in *Proc. of the 41st IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2016, pp. 296–300.

[9] A. Gkiokas and V. Katsouros, "Convolutional neural networks for real-time beat tracking: A dancing robot application," in *Proc. of the 18th Intl. Society for Music Information Retrieval Conf.*, 2017, pp. 286–293.

[10] R. Vogl, M. Dorfer, G. Widmer, and P. Knees, "Drum transcription via joint beat and drum modeling using convolutional recurrent neural networks," in *Proc. of the 18th Intl. Society for Music Information Retrieval Conf.*, 2017, pp. 150–157.

[11] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "Analysis of common design choices in deep learning systems for downbeat tracking," in *Proc. of the 19th Intl. Society for Music Information Retrieval Conf.*, 2018, pp. 106–112.

[12] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[13] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.

[14] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *Proc. of the 39th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2014, pp. 6979–6983.

[15] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," in *Proc. of the 4th Intl. Conf. on Learning Representations*, 2016.

[16] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd Intl. Conf. for Learning Representations*, 2015.

[18] F. Krebs, S. Böck, and G. Widmer, "An Efficient State Space Model for Joint Tempo and Meter Tracking," in *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, 2015, pp. 72–78.

[19] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," Centre for Digital Music, Queen Mary University of London, Tech. Rep. C4DM-TR-09-06, 2009.

[20] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[21] U. Marchand and G. Peeters, "Swing ratio estimation," in *Proc. of the 18th Intl. Conf. on Digital Audio Effects*, 2015, pp. 423–428.

[22] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, p. 1832–1844, 2006.

[23] F. Krebs, S. Böck, and G. Widmer, "Rhythmic pattern modeling for beat and downbeat tracking in musical audio," in *Proc. of the 14th Intl. Society for Music Information Retrieval Conf.*, 2013, pp. 227–232.

[24] S. Hainsworth and M. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2385–2395, 2004.

[25] F. Gouyon, "A computational approach to rhythm description — audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing," Ph.D. dissertation, Universitat Pompeu Fabra, 2005.

[26] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

Towards user-informed beat tracking of musical audio

António Sá Pinto[1,2] and Matthew E. P. Davies[1] *

[1] INESC TEC, Sound and Music Computing Group, Porto, Portugal
[2] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal
{antonio.s.pinto,matthew.davies}@inesctec.pt

Abstract. We explore the task of computational beat tracking for musical audio signals from the perspective of putting an end-user directly in the processing loop. Unlike existing "semi-automatic" approaches for beat tracking, where users may select from among several possible outputs to determine the one that best suits their aims, in our approach we examine how high-level user input could guide the manner in which the analysis is performed. More specifically, we focus on the perceptual difficulty of tapping the beat, which has previously been associated with the musical properties of expressive timing and slow tempo. Since musical examples with these properties have been shown to be poorly addressed even by state of the art approaches to beat tracking, we re-parameterise an existing deep learning based approach to enable it to more reliably track highly expressive music. In a small-scale listening experiment we highlight two principal trends: i) that users are able to consistently disambiguate musical examples which are easy to tap to and those which are not; and in turn ii) that users preferred the beat tracking output of an expressive-parameterised system to the default parameterisation for highly expressive musical excerpts.

Keywords: Beat Tracking, Expressive Timing, User Input

1 Introduction and Motivation

While the task of computational beat tracking is relatively straightforward to define – its aim being to replicate the innate human ability to synchronise with a musical stimulus by tapping a foot along with the beat – it remains a complex and unsolved task within the music information retrieval (MIR) community. Scientific progress in MIR tasks is most often demonstrated through improved accuracy scores when compared with existing state of the art methods [18]. At the core of this comparison rest two fundamental tenets: the (annotated) data upon which the algorithms are evaluated, and the evaluation method(s) used to measure performance. In the case of beat tracking, both the tasks of annotating

* António Sá Pinto and Matthew E. P. Davies are supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the grant SFRH/BD/120383/2016 and the project IF/01566/2015.

datasets of musical material and measuring performance are non-trivial [6]. By its very nature, the concept of beat perception – how an individual perceives the beat in a piece of music – is highly subjective [15]. When tapping the beat, listeners may agree over the phase, but disagree over the tempo or preferred metrical level – with one tapping, *e.g.*, twice as fast as another, or alternatively, they may agree over the tempo, but tap in anti-phase. This inherent ambiguity led to the prevalence of multiple hypotheses of the beat, which can arise at the point of annotation, but more commonly appear during evaluation where different interpretations of ground truth annotations are obtained via interpolation or sub-sampling. In this way, a wide net can be cast in order not to punish beat tracking algorithms which fail to precisely match the annotated metrical level or phase of the beats; with this coming at the expense that some unlikely beat outputs may inadvertently be deemed accurate. Following this evaluation strategy, the performance of the state of the art is now in the order of 90% on existing datasets [3, 4] comprised primarily of pop, rock and electronic dance music. However, performance on more challenging material [10] is considerably lower, with factors such as expressive timing (*i.e.*, the timing variability that characterises a human performance, in opposition to a metronomic or perfectly timed rendition [7]), recording quality, slow tempo and metre changes among several identified challenging properties.

Although beat tracking has garnered much attention in the MIR community, it is often treated as an element in a more complex processing pipeline which provides access to "musical time", or simply evaluated based on how well it can predict ground truth annotations. Yet, within the emerging domain of creative-MIR [16, 11] the extraction of the beat can play a critical role in musically-responsive and interactive systems [13]. A fundamental difference of applying beat tracking in a creative application scenario is that there is a specific end-user who wishes to directly employ the music analysis and thus has very high expectations in terms of its performance [1]. To this end, obtaining high mean accuracy scores across some existing databases is of lower value than knowing "*Can the beats be accurately extracted (as I want them) for this specific piece of music?*". Furthermore, we must also be aware that accuracy scores themselves may not be informative about "true" underlying performance [17, 6].

Of course, a user-specific beat annotation can be obtained without any beat tracking algorithm, by manually annotating the desired beat locations. However, manually annotating beat locations is a laborious procedure even for skilled annotators [10]. An alternative is to leverage multiple beat interpretations from a beat tracking algorithm, and then provide users with a range of solutions to choose from [8]. However, even with a large number of interpretations (which may be non-trivial and time-consuming to rank) there is no guarantee that the end-user's desired result will be present, especially if the alternative interpretations are generated in a deterministic manner from a single beat tracking output, *e.g.*, by interpolation or sub-sampling.

In this paper, we propose an alternative formulation which allows an end-user to drive how the beat tracking is undertaken. Our goal is to enable the

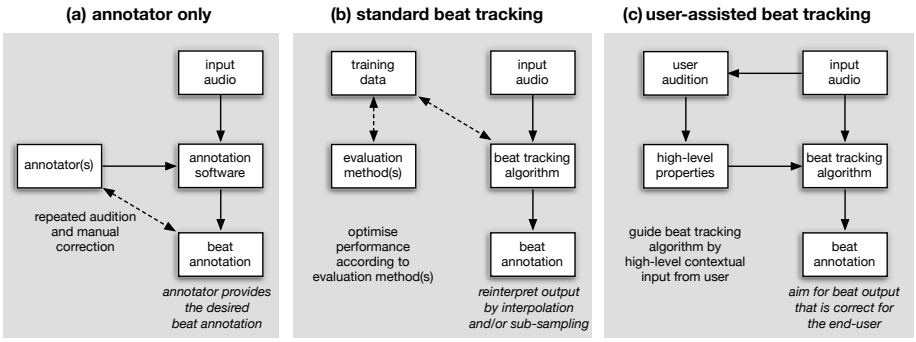


Fig. 1. Overview of different approaches to obtaining a desired beat annotation. (a) The user annotates the beat positions. (b) A beat tracking algorithm is used – whose performance has been optimised on annotated datasets. (c) Our proposed approach, where user input guides the beat tracking.

user to rapidly arrive at the beat annotation suitable for their purposes with a minimal amount of interaction. Put another way, we envisage an approach to beat tracking where high-level contextual knowledge about a specific musical signal can be given by the user and reliably interpreted by the algorithm, without the need for extensive model training on annotated datasets, as shown in Fig. 1. In this sense, we put aside the concept of "universal" beat tracking models which target equal performance irrespective of the musical input signal, in favour of the more realistic goal of identifying different classes of the beat tracking problem, which require different beat tracking strategies. While the end goal of retrieving beat locations may be the same for fast-paced techno music and highly expressive classical guitar recordings, the assumptions about what constitutes the beat, and how this can be extracted from audio signals are not. Conversely, constraints should not be placed on what musical content can be creatively re-purposed based on the limitations of MIR algorithms.

The long term challenges of our approach are as follows: i) determining a low-dimensional parameterisation of the beat tracking space within which diverse, accurate solutions can be found in order to match different beat tracking conditions; ii) exposing these dimensions to end-users in a way that they can be easily understood; iii) providing an interpretable and understandable mapping between the user-input and the resulting beat annotation via the beat tracking algorithm; and finally iv) measuring the level of engagement among end-users who actively participate in the analysis of music signals.

Concerning the dimensions of beat tracking, it is well-understood that music of approximately constant (medium) tempo, with strong percussive content (*e.g.*, pop, rock music) is straightforward to track. Beat tracking difficulty (both for computational approaches and human tappers) can be due to musical reasons and signal-based properties [9, 10]. While it is somewhat nonsensical to consider a piece of music with "opposite" properties to the most straightforward case, it has

been shown empirically that highly expressive music, without clear percussive content, is not well analysed even by the state of the art in beat tracking [10, 4]. Successful tracking of such pieces should, in principle, require input features which can be effective in the absence of percussion and a tracking model which can rapidly adapt to expressive tempo variation. While recent work [3] sought to develop multiple beat tracking models, these were separately trained at the level of different databases rather than according to musical beat tracking conditions.

In our approach, we reexamine the functionality of the current state of the art in beat tracking, *i.e.*, the recurrent neural network approach of Böck et al. [4]. In particular, we devise a means to re-parameterise it so that it is adapted for highly expressive music. Based on an analysis of existing annotated datasets, we identify a set of musical stimuli we consider typical of highly challenging conditions, together with a parallel set of "easier" examples. We then conduct a small-scale listening experiment where participants are first asked to rate the perceptual difficulty of tapping the beat, and subsequently to rate the subjective quality of beat annotations given by the expressive parameterisation vs the default version. Our results indicate that listeners are able to distinguish easier from more challenging cases, and furthermore that they preferred the beat tracking output of the expressive-parameterised system to the default parameterisation for the highly expressive musical excerpts. In this sense, we seek to use the assessment of perceptual difficulty of tapping as a means to drive the manner in which the beats can be extracted from audio signals towards the concept of user-informed beat tracking. To complement our analysis, we explore the objective evaluation of the beat tracking model with both parameterisations.

The remainder of this paper is structured as follows. In Section 2 we detail the adaption of the beat tracking followed by the design of a small-scale listening experiment in Section 3. This is followed by results and discussion in Section 4, and conclusions in Section 5.

2 Beat Tracking System Adaptation

Within this work our goal is to include user input to drive how music signal analysis is conducted. We hypothesise that high-level contextual information which may be straightforward for human listeners to determine can provide a means to guide how the music signal analysis is conducted. For beat tracking, we established in Section 1 that for straightforward musical cases, the current state of the art [4] is highly effective. Therefore, in order to provide an improvement over the state of the art, we must consider the conditions in which it is less effective, in particular those displaying expressive timing. To this end, we first summarise the main functionality of the beat tracking approach of Böck et al., after which we detail how we adapt it.

The approach of Böck et al. [4] uses deep learning and is freely available within the madmom library [2]. The core of the beat tracking model is a recurrent neural network (RNN) which has been trained on a wide range of annotated beat tracking datasets to predict a beat activation function which exhibits peaks at

likely beat locations. To obtain an output beat sequence, the beat activation function given by the RNN is post-processed by a dynamic Bayesian network (DBN) which is approximated by a hidden Markov model [14].

While it would be possible to retain this model from scratch on challenging data, this has been partially addressed in the earlier multi-model approach of Böck et al. [3]. Instead, we reflect on the latter part of the beat tracking pipeline, namely how to obtain the beat annotation from the beat activation function. To this end, we address three DBN parameters: i) the minimum tempo in beats per minute (BPM); ii) the maximum tempo; and iii) the so-called "transition-λ" parameter which controls the flexibility of the DBN to deviate from a constant tempo[3]. Through iterative experimentation, including both objective evaluation on existing datasets and subjective assessment of the quality of the beat tracking output, we devised a new set of expressiveness-oriented parameters, which are shown, along with the default values in Table 1. More specifically, we first undertake a grid search across these three parameters on a subset of musical examples from existing annotated datasets for which the state of the art RNN is deemed to perform poorly, *i.e.*, by having an information gain lower than 1.5 bits [19]. An informal subjective assessment was then used to confirm that reliable beat annotations could be obtained from the expressive parameterisation.

Table 1. Overview of default and expressive-adapted parameters.

Parameter	**Default**	**Expressive**
Minimum Tempo (BPM)	55	35
Maximum Tempo (BPM)	215	135
Transition-λ (unitless)	100	10

As shown in Table 1, the main changes for the expressive model are a shift towards a slower range of allowed tempi (following evidence about the greater difficulty of tapping to slower pieces of music [5]), together with a lower value for the transition-λ. While the global effect of the transition-λ was studied by Krebs et al. [14], their goal was to find an optimal value across a wide range of musical examples. Here, our focus is on highly expressive music and therefore we do not need a more general solution. Indeed, the role of the expressive model is to function in precisely the cases where the default approach can not.

3 Experimental Design

Within this paper, we posit that high-level user-input can lead to improved beat annotation over using existing state of the art beat tracking algorithms in a

[3] the probability of tempo changes varies exponentially with the negative of the "transition-λ", thus higher values of this parameter favour constant tempo from one beat to the next one [14].

"blind" manner. In order to test this in a rigorous way, we would need to build an interactive beat tracking system including a user interface, and conduct a user study in which users could select their own input material for evaluation. However, doing so would require understanding which high-level properties to expose and how to meaningfully interpret them within the beat tracking system. To the best of our knowledge, no such experiment has yet been conducted, thus in order to gain some initial insight into this problem, we conducted a small-scale online listening experiment, which is split into two parts: **Part A** to assess the perceptual difficulty of tapping the beat, and **Part B** to assess the subjective quality of beat annotations made using the default parameterisation of the state of the art beat tracking system versus our proposed expressive parameterisation.

We use **Part A** as a means to simulate one potential aspect of high-level context which an end-user could provide: in this case, a choice over whether the piece of music is easy or difficult to tap along to (where difficulty is largely driven by the presence of expressive timing). Given this choice, **Part B** is used as the means for the end-user to rate the quality of the beat annotation when the beat tracking system has been parameterised according to their choice. In this sense, if a user rates the piece as "easy", we would provide the default output of the system, and if they rate it as "hard" we provide the annotation from the expressive parameterisation. However, for the purposes of our listening experiment, all experimental conditions are rated by all participants, thus the link between **Part A** and **Part B** is not explicit.

3.1 Part A

In the first part of our experiment, we used a set of 8 short music excerpts (each 15 s in duration) which were split equally among two categories: i) "easy" cases with near constant tempo in 4/4 time, with percussive content, and without highly syncopated rhythmic patterns; and ii) "hard" cases typified by the presence of high tempo variation and minimal use of percussion. The musical excerpts were drawn from existing public and private beat tracking datasets, and all were normalised to -3 dB.

We asked the participants to listen to the musical excerpts and to spontaneously tap along using the computer keyboard at what they considered the most salient beat. Due to the challenges of recording precise time stamps without dedicated signal acquisition hardware (*e.g.*, at the very least, a MIDI input device) the tap times of the participants were not recorded, however this was not disclosed. We then asked the participants to rate the difficulty they felt when trying to tap the beat, according to the following four options:

- Low - *I could easily tap the beat, almost without concentrating*
- Medium - *It wasn't easy, but with some concentration, I could adequately tap the beat*
- High - *I had to concentrate very hard to try to tap the beat*
- Extremely high - *I was not able to tap the beat at all.*

Our hypothesis for **Part A** is that participants should consistently rate those drawn from the "easy" set as having Low or Medium difficulty, whereas those from the "hard" should be rated with High or Extremely High difficulty.

3.2 Part B

Having completed **Part A**, participants then proceeded to **Part B** in which they were asked to judge the subjective quality of beat annotations (rendered as short 1 kHz pulses) mixed with the musical excerpts. The same set of musical excerpts from **Part A** were used, but they were annotated in three different ways: i) using the *default* parameterisation of the Böck et al. RNN approach from the madmom library [2]; ii) using our proposed *expressive* parameterisation (as in Table 1); and iii) a control condition using a completely *deterministic* beat annotation, *i.e.*, beat times at precise 500 ms intervals without any attempt to track the beat of the music. In total, this created a set of $8 \times 3 = 24$ musical excerpts to be rated, for which participants were asked to: *Rate the overall quality of how well the beat sequence corresponds to the beat of the music.*

For this question, a 5-point Likert-type item was used with (1) on the left hand side corresponding to "Not at all" and (5) corresponding to "Entirely" on the right hand side. Our hypothesis for **Part B** was that for the "hard" excerpts, the annotations of the expressively-parameterised beat tracker would be preferred to those of the default approach, and for all musical excerpts that the deterministic condition would be rated the lowest in terms of subjective quality.

3.3 Implementation

The experiment was built using HTML5 and Node.js and run online within a web browser, where participants were recruited from the student body of the University of Porto and the research network of the Sound and Music Computing Group at INESC TEC. Within the experimental instructions, all participants were required to give their informed consent to participate, with the understanding that any data collected would be handled in an anonymous fashion and that they were free to withdraw at any time without penalty (and without their partial responses being recorded). Participants were asked to provide basic information for statistical purposes: sex, age, their level of expertise as a musician, and experience in music production.

All participants were encouraged to take the experiment in a quiet environment using high quality headphones or loudspeakers, and before starting, they were given the opportunity to set the playback volume to a comfortable level. Prior to the start of each main part of the experiment, the participants undertook a compulsory training phase in order to familiarise themselves with the questions. To prevent order effects, each participant was presented with the musical excerpts in a different random order. In total, the test took around 30 minutes to complete.

4 Results and Discussion

4.1 Listening Experiment

A total of 10 listeners (mean age: 31, age range: 23–43) participated in the listening test, 9 of whom self-reported amateur or professional musical proficiency.

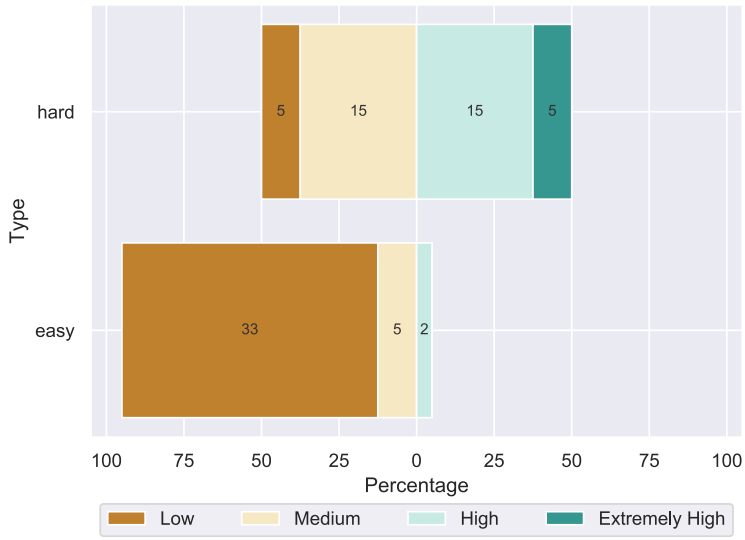


Fig. 2. Subjective ratings of the difficulty of beat tapping.

For **Part A**, we obtained 40 ratings for each stimuli group "easy" and "hard", according to the frequency distribution shown in Fig. 2. The most frequent rating for the first group was "low" (82.5%), followed by the "medium" rating (12.5%). For the "hard" group, a symmetrical rating was obtained: the adjacent ratings "medium" and "high" (37.5% each), complemented by the more extreme ratings "low" and "extremely high" (12.5% each). A Mann-Whitney test showed that there was a statistically significant difference between the ratings for both groups, with $p < 0.001$.

From these results we interpret that there was greater consistency in classifying the "easy" excerpts as having low difficulty, with only two excerpts rated above "medium", than for the "hard" excerpts which covered the entire rating scale from low to extremely difficult, albeit with the majority of ratings being for medium or high difficulty. We interpret this greater variability in the rating of difficulty of tapping to be the product of two properties of the participants: their expertise in musical performance and/or their familiarity with the pieces. Moreover, we can observe a minor separation between the understanding of the

perceptual difficulty in tapping on the part of the participant and the presence of expressive timing in the musical excerpts; that experienced listeners may not have difficulty in tapping along with a piece of expressive music for which they knew well. Thus, for expert listeners it may be more reasonable to ask a direct question related to the presence of expressive timing, while the question of difficulty may be more appropriate for non-expert listeners who might lack familiarity with the necessary musical terminology.

For **Part B**, we again make the distinction between the ratings of the "easy" and the "hard" excerpts. A Kruskal-Wallis H test showed that there was a statistically significant difference between the 3 models (*expressive*, *default* and *deterministic*): $\chi^2(2) = 87.96$, $p < 0.001$ for "easy" excerpts, $\chi^2(2) = 70.71$, $p < 0.001$ for "hard" excerpts. A post-hoc analysis performed with the Dunn test with Bonferroni correction showed that all the differences were statistically significant with $p < 0.001/3$ (except for the pair *default–expressive* under the "easy" stimuli, for which identical ratings were obtained). A descriptive summary of the ratings (boxplot with scores overlaid) for each type of stimuli, and under the three beat annotation conditions are shown in Fig. 3.

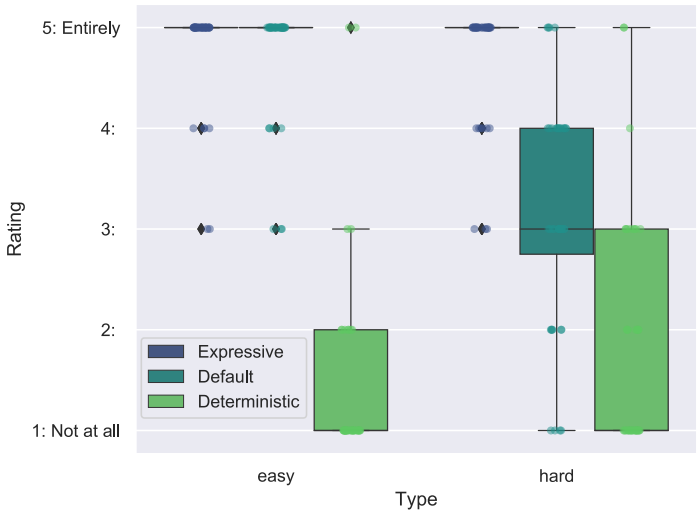


Fig. 3. Subjective ratings of the quality of the beat annotations.

The main results from Part B are as **follows. For** the "easy" excerpts there is no difference in performance for the *default* and *expressive* parameterisations of the beat tracking model, both of which are rated with high scores indicating high quality beat annotations from both systems. We contrast this with the ratings of the *deterministic* output (which should bear no meaningful relationship to the music) and which are rated toward the lower end of the scale. From these

results we can infer that the participants were easily able to distinguish accurate beat annotations and entirely inaccurate annotations, which is consistent with the Beat Alignment Test [12]. Concerning the ability of the expressively parameterised model to achieve such high ratings, we believe that this was due to very clear information concerning the beat in the beat activation functions from the RNN.

Conversely, the ratings of the "hard" excerpts show a different picture. Here, the ratings of the expressively parameterised model are similar to the "easy" excerpts, but the ratings of the *default* model [2] are noticeably lower. This suggests that the participants, in spite of their reported higher perceptual difficulty in tapping the beat, were able to reliably identify the accurate beat predictions of the *expressive* model over those of the *default* model. It is noteworthy that the ratings of the *deterministic* approach are moderately higher for the "hard" excerpts compared to the "easy" excerpts. Given the small number of samples and participants for this experiment, it is hard to draw strong conclusions about this difference, but for highly expressive pieces, the *deterministic* beats may have inadvertently aligned with the music in brief periods compared to the "easy" excerpts, which may have been unambiguously unrelated.

4.2 Beat Tracking Accuracy

In addition to reporting on the listening experiment whose focus is on subjective ratings of beat tracking, we also examine the difference in objective performance of using the *default* and *expressive* parameterisations of the beat tracking model. Given the focus on challenging excerpts for beat tracking, we focus on the SMC dataset [10]. It contains 217 excerpts, each of 40 s in duration. Following the evaluation methods described in [6] we select a common subset: F-measure, CMLc, CMLt, AMLc, AMLt, and the Information Gain (D) to assess performance. In Table 2, we show the recorded accuracy on this dataset for both the default and expressive parameterisations. Note, for the default model we use the version in the madmom library [2] which has been exposed to this material during training, hence the accuracy scores are slightly higher than those in [4] where cross fold validation was used. In addition to showing the performance of each parameterisation we also show the theoretical upper limit achievable by making a perfect choice (by a hypothetical end-user) among the two parameterisations.

Table 2. Overview of beat tracking performance on the SMC dataset [10] comparing the default and expressive parameters together with upper limit on performance.

	F-measure	CMLc	CMLt	AMLc	AMLt	D
Default[2]	0.563	0.350	0.472	0.459	0.629	1.586
Expressive	0.540	0.306	0.410	0.427	0.565	1.653
Optimal Choice	0.624	0.456	0.611	0.545	0.703	1.830

From Table 2, we see that for all the evaluation methods, with the exception of the Information Gain (D), the default parameterisation outperforms the expressive one. This is an expected result since the dataset is not entirely comprised of highly expressive musical material. We consider the more important result to be the potential for our *expressive* parameterisation to track those excerpts for which the *default* approach fails. To this end, the increase of approximately 10% points across each of the evaluation methods demonstrates how these two different parameterisations can provide greater coverage of the dataset. It also implies that training a binary classifier to choose between expressive and non-expressive pieces would be a promising area for future work.

5 Conclusions

In this paper we have sought to open the discussion about the potential for user-input to drive how MIR analysis is performed. Within the context of beat tracking, we have demonstrated that it is possible to reparameterise an existing state-of-the-art approach to provide better beat annotations for highly expressive music, and furthermore, that the ability to choose between the default and expressive parameterisation can provide significant improvements on very challenging beat tracking material. We emphasise that the benefit of the expressive model was achieved without the need for any retraining of the RNN architecture, but that the improvement was obtained by reparameterisation of the DBN tracking model.

To obtain some insight into how user input could be used for beat tracking, we simulated a scenario where user decisions about perceptual difficulty of tapping could be translated into the use of a parameterisation for expressive musical excerpts. We speculate that listener expertise as well as familiarity may play a role in lowering the perceived difficulty of otherwise challenging expressive pieces. Our intention is to further investigate the parameters which can be exposed to end-users, and whether different properties may exist for expert compared to non-expert users. Despite the statistical significance of our results, we recognise the small-scale nature of the listening experiment, and we intend to expand both the number of musical excerpts uses as well as targeting a larger group of participants to gain deeper insight into the types of user groups which may emerge. Towards our long-term goal, we will undertake an user study not only to understand the role of beat tracking for creative MIR, but also to assess the level of engagement when end-users are active participants who guide the analysis.

References

1. K. Andersen and P. Knees. Conversations with Expert Users in Music Retrieval and Research Challenges for Creative MIR. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.*, pages 122–128, 2016.
2. S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In *Proc. of the 2016 ACM Multimedia Conf.*, pages 1174–1178, 2016.

3. S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proc. of the 15th Intl. Society for Music Information Retrieval Conf.*, pages 603–608, 2014.
4. S. Böck, F. Krebs, and G. Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.*, pages 255–261, 2016.
5. R. Bååth and G. Madison. The subjective difficulty of tapping to a slow beat. In *Proc. of the 12th Intl. Conf. on Music Perception and Cognition*, pages 82–55, 2012.
6. M. E. P. Davies and S. Böck. Evaluating the evaluation measures for beat tracking. In *Proc. of the 15th Intl. Society for Music Information Retrieval Conf.*, pages 637–642, 2014.
7. P. Desain and H. Honing. Does expressive timing in music performance scale proportionally with tempo? *Psychological Research*, 56(4):285–292, 1994.
8. M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano. Songle: A Web Service for Active Music Listening Improved by User Contributions. In *Proc. of the 12th Intl. Society for Music Information Retrieval Conf.*, pages 311–316, 2011.
9. P. Grosche, M. Müller, and C. Sapp. What makes beat tracking difficult? a case study on chopin mazurkas. In *Proc. of the 11th Intl. Society for Music Information Retrieval Conf.*, pages 649–654, 2010.
10. A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 20(9):2539–2460, 2012.
11. E. J. Humphrey, D. Turnbull, and T. Collins. A brief review of creative MIR. In *Late-breaking demo session of the 14th Intl. Society for Music Information Retrieval Conf.*, 2013.
12. J. R. Iversen and A. D. Patel. The Beat Alignment Test (BAT): Surveying beat processing abilities in the general population. In *Proc. of the 10th Intl. Conf. on Music Perception and Cognition*, pages 465–468, 2010.
13. C. T. Jin, M. E. P. Davies, and P. Campisi. Embedded Systems Feel the Beat in New Orleans: Highlights from the IEEE Signal Processing Cup 2017 Student Competition [SP Competitions]. *IEEE Signal Processing Magazine*, 34(4):143–170, 2017.
14. F. Krebs, S. Böck, and G. Widmer. An efficient state space model for joint tempo and meter tracking. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, pages 72–78, 2015.
15. D. Moelants and M. McKinney. Tempo perception and musical content: what makes a piece fast, slow or temporally ambiguous? In *Proc. of the 8th Intl. Conf. on Music Perception and Cognition*, pages 558–562, 2004.
16. X. Serra et al. Roadmap for music information research, 2013. Creative Commons BY-NC-ND 3.0 license, ISBN: 978-2-9540351-1-6.
17. B. L. Sturm. Classification accuracy is not enough. *Journal of Intelligent Information Systems*, 41(3):371–406, 2013.
18. J. Urbano, M. Schedl, and X. Serra. Evaluation in Music Information Retrieval. *Journal of Intelligent Information Systems*, 41(3):345–369, 2013.
19. J. R. Zapata, A. Holzapfel, M. E. P. Davies, J. L. Oliveira, and F. Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *Proc. of the 13th Intl. Society for Music Information Retrieval Conf.*, pages 157–162, 2012.

MULTI-TASK LEARNING OF TEMPO AND BEAT: LEARNING ONE TO IMPROVE THE OTHER

Sebastian Böck[1,3] **Matthew E.P. Davies**[2] **Peter Knees**[3]
[1] Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria
[2] INESC TEC, Porto, Portugal [3] TU Wien, Vienna, Austria
sebastian.boeck@ofai.at

ABSTRACT

We propose a multi-task learning approach for simultaneous tempo estimation and beat tracking of musical audio. The system shows state-of-the-art performance for both tasks on a wide range of data, but has another fundamental advantage: due to its multi-task nature, it is not only able to exploit the mutual information of both tasks by learning a common, shared representation, but can also improve one by learning only from the other. The multi-task learning is achieved by globally aggregating the skip connections of a beat tracking system built around temporal convolutional networks, and feeding them into a tempo classification layer. The benefit of this approach is investigated by the inclusion of training data for which tempo-only annotations are available, and which is shown to provide improvements in beat tracking accuracy.

1. INTRODUCTION

By definition, the music analysis tasks of tempo estimation and beat tracking are highly interconnected. Considering the goal of a beat tracking system is to produce a sequence of time instants that reflect how a human listener might tap their foot in time to a piece of music, we understand the tempo as the rate at which these beats occur, as measured in beats per minute (BPM). With the exception of a specific class of musical recordings which are both perfectly quantised (i.e. adhering strictly to a fixed metronome), and which begin precisely at the onset of a beat, e.g. drum loops, tempo information alone is insufficient to derive the beats since it provides no information about phase. In practice, a more flexible and musically realistic approach to beat tracking is required to contend with deviations from purely isochronous beat sequences without a trivial phase component. These deviations can take the form of continuous changes in tempo and/or timing which are common in expressive musical performances, more abrupt "step" changes in tempo, or short pauses after which a previously established tempo is resumed [21]. The presence and extent of these deviations from isochrony have been identified as contributing to the difficulty of musical examples for computational beat tracking [14] as well as for human annotators annotating ground truth [27].

When reflecting on the history of computational approaches for beat tracking, we consider that the role and usage of data has fundamentally changed. For the earliest work in beat tracking in the 1990s [18, 37], annotated data was scarce. By the mid-to-late 2000s, several beat tracking datasets (both public and private) came into use [12, 19, 20, 22, 24, 29] and were widely adopted as the primary means for reporting beat tracking performance. Even allowing for parameter optimisation or some training to maximise the performance of beat tracking algorithms on given datasets, a closed loop (in a strict end-to-end sense) did not exist between annotated data and beat tracking algorithms until the advent of deep neural network (DNN) approaches [7]. Both the high learning power and explicit use of annotations of DNN approaches led to a significant leap in the state of the art.

Similarly, tempo induction algorithms formerly tried to identify the main periodicity of musical accent features, such as band-passed signals, discrete onsets or a continuous detection function by means of auto-correlation [1, 13, 36], resonating comb filters [29, 37] or Fourier analysis [9], and available data was only used to evaluate the algorithms. The first attempts to learn something meaningful from data for tempo estimation sought to devise ways to choose among multiple tempo hypotheses [15, 16, 26, 38, 45] or to predict the perceptual tempo [35]. Only recently, DNN approaches have been used to infer tempo directly from spectral features [40].

At the present time, DNN approaches are highly prevalent among music analysis and generation research within the music information retrieval (MIR) community, and thus access to large amounts of high-quality annotated data is of paramount importance for the development and training of new models. For beat tracking, the hand annotation of beat locations is particularly arduous due to the need to make several hundred temporally-dependent annotations per full piece of music, and the work-load only increases in the presence of challenging musical and signal conditions [27]. By contrast, global tempo annotation, while still dependent on some approximate beat marking, can typically be created with far less effort. As a result,

there is a far greater amount of tempo annotated data available than for beat tracking.

Our motivation is therefore towards a new approach for beat tracking which can be trained not only on beat annotations but also from tempo-only annotated data. We formulate this as a multi-task learning problem [8] for simultaneous tempo estimation and beat tracking. Our hypothesis is that due to the multi-task nature, we can not only exploit the mutual information of both tasks by learning a common, shared representation, but also improve one by learning only from the other.

We implement our multi-task approach by extending a recent beat tracking system [11] built around temporal convolutional networks (TCNs) [2, 44]. The primary addition in this paper takes the form of globally aggregating the skip connections of the TCN and feeding them into a tempo classification layer. A graphical overview of the inputs and outputs of our system is shown in Figure 1, with details of the architecture in Figure 2.

We evaluate our proposed multi-task system on a wide range of existing beat- and tempo-annotated datasets and compare performance against reference systems in both tasks. Our results demonstrate that the multi-task formulation achieves state-of-the-art performance in both tempo estimation and beat tracking. The most notable increase in performance occurs on a dataset where the network has been trained on tempo labels but whose beat annotations remain totally unseen by the network.

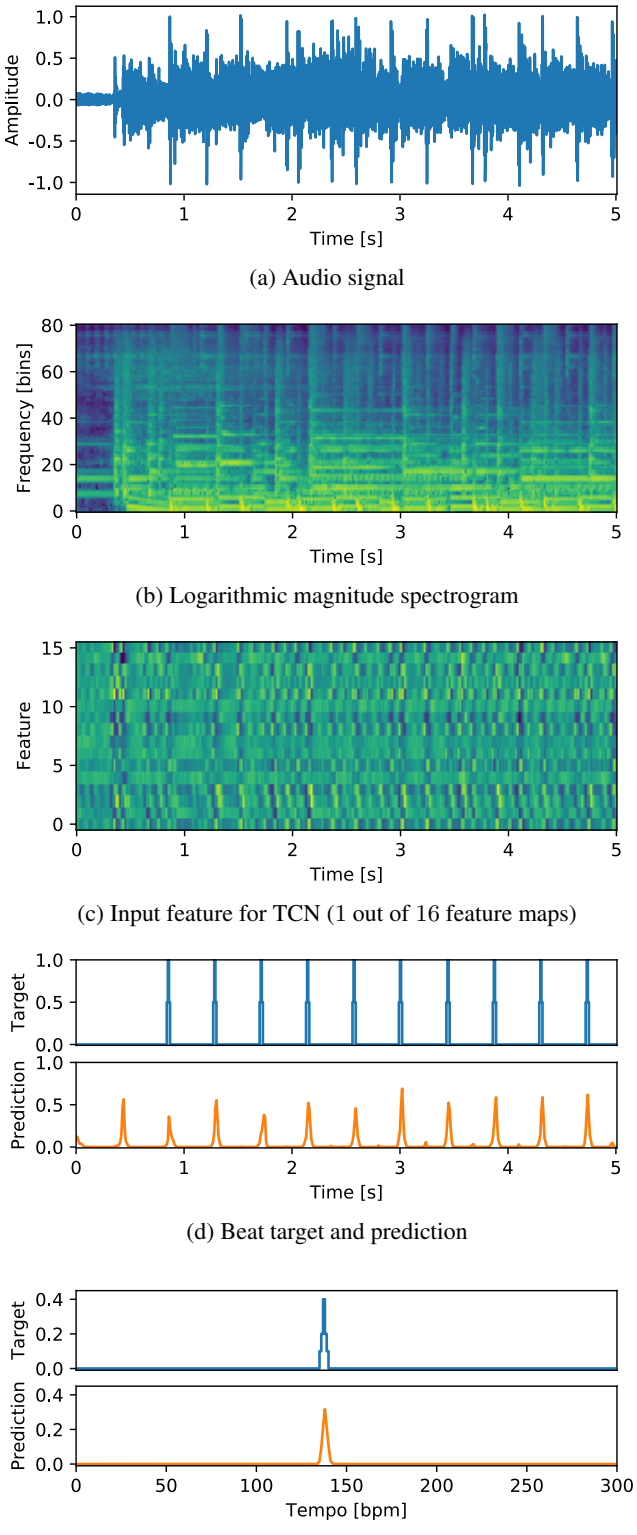
The remainder of this paper is structured as follows. In Section 2 we provide an overview of the existing beat tracking approach and then detail our multi-task formulation. In Section 3 we conduct a rigorous evaluation of beat tracking and tempo estimation. Finally, in Section 4 we discuss the context of the results and propose areas for future work.

2. APPROACH

In this section, we provide an overview of the beat tracking system [11] around which our multi-task learning approach is formulated. Following this, we describe the extension for multi-task learning via the inclusion of an additional output layer which performs tempo classification.

2.1 Beat Tracking Approach

The underlying beat tracking approach is inspired by two well-known deep learning methods: i) the *WaveNet* model [44] which uses dilated convolutions for generative audio synthesis by learning directly on raw audio waveforms, and ii) the current state of the art in musical audio beat tracking [4, 6], which uses a bi-directional long short-term memory (BLSTM) recurrent architecture. Based on the work of Bai et al. [2], who demonstrated improved performance of TCNs over recurrent architectures for numerous sequential data analysis and classification tasks, we developed a TCN approach for musical audio beat tracking [11] which, at a high-level, addressed the substitution of the BLSTM in [4, 6] with a TCN. However, since the



(a) Audio signal

(b) Logarithmic magnitude spectrogram

(c) Input feature for TCN (1 out of 16 feature maps)

(d) Beat target and prediction

(e) Tempo target and prediction

Figure 1: Signal flow of a 5 second audio excerpt through the proposed multi-task system. From the time domain signal (a), a logarithmic magnitude spectrogram is computed (b). This input representation is processed by intermediate convolutional and max pooling layers to obtain a single 16-dimensional feature (c), which is fed into the TCN. Both targets and predictions for beats and tempo are shown in (d) and (e), respectively.

TCN from *WaveNet* is both causal and operates on raw audio, several modifications were required, which are summarised below.

Instead of using raw audio as input, the dilated convolutions are performed on a highly sub-sampled low-dimensional feature representation (cf. Figure 1c). This 16-dimensional feature vector is derived by applying multiple convolution and max pooling operations to a log magnitude spectrogram of the input audio signal. The spectrogram is computed with a window and FFT size of 2048 samples, a hop size of 441 samples (i.e. 100 frames per second for audio sampled at 44100 Hz), and filtered with a bank of overlapping triangular filters with 12 bands per octave covering a frequency range of 30 to $17,000$ Hz (cf. Figure 1b). Alternating convolutional and max pooling layers are applied to slices of 5 frames in length to reduce the dimensionality both in time and frequency to a single dimension. The convolutional layers contain 16 filters each, with kernel sizes of 3×3 for the first two, and 1×8 for the last layer. The intermediate max pooling layers apply pooling only in the frequency direction over 3 frequency bins. A dropout [42] rate of 0.1 is used with the exponential linear unit (*ELU*) [10] as activation function.

The main TCN component from *WaveNet* was modified to operate non-causally, meaning that, for any time frame of the input representation, the dilated convolutions extend in both directions (i.e. back to the past and forward to the future). This provides a receptive field which is centred on the time frame in question, rather than directed solely towards the past.

In terms of the parameterisation of the TCN approach we used 11 layers with 16 1-dimensional filters of size 5 and geometrically spaced dilations ranging from 2^0 up to 2^{10} time frames. The resulting receptive field is ~ 81.5 seconds. We applied spatial dropout with rate 0.1 and used the *ELU* activation function instead of the gated activations of *WaveNet*. As output we used a single *sigmoid* unit. In order to obtain a final beat tracking output, the beat activation function produced by the network was passed to a dynamic Bayesian network, approximated by a hidden Markov model, from [31]. For further details on the TCN approach for beat tracking, see [11].

In this work we slightly changed the architecture of [11] by adding another 1×1 convolution layer with 16 filters into the residual path of the TCN layers (cf. Figure 2). We found that this layer helped to increase tempo estimation performance.

2.2 Multi-Task Extension

We extend this beat tracking system to be able to estimate the tempo of a musical piece by adding a second output branch to the network. As output, a classification layer with linear spacing as in [40] is used. It has 300 units, representing a tempo range from 0 (indicating that the piece has no tempo) up to 300 BPM. This additional output allows for multi-task learning of the whole system, the details of which are outlined in Figure 2. In order to be able to process input sequences of variable length, global average pooling (over time) is used to aggregate the features for the tempo classification layer.

While it is possible to feed the output of the TCN (or indeed the output of any other sequential beat tracking model) directly to the tempo classification layer, in practice we found that using a beat activation function led to reasonable "coarse" tempo estimation performance (i.e. determining whether a musical piece is either fast or slow), but lacked absolute precision. However, utilising skip connections of the TCN boosted tempo estimation accuracy considerably. Our intuition is that this way the subtleties of the intermediate representation of the dilated convolutions (which represent different time scales) are preserved and can be better exploited.

In the original *WaveNet* [44], skip connections were used to speed up convergence and enable training of deep models. Since the TCN beat tracking system [11] has only 11 layers, skip connections were not needed to successfully train a model and thus were not utilised. In this work, we branch off the skip connections at the same location as in *WaveNet* (i.e. from the 1×1 convolutions inside the TCN layers), but use them solely for the tempo branch of the network (cf. Figure 2).

We aggregate the skip connections of the individual layers by summation. Since the 1×1 convolutions have 16 filters each, this results in a single 16-dimensional feature vector for classification. Compared to concatenating the skip connections, this low-dimensional input to the tempo classification layer reliably prevents over-fitting to the training data. We apply dropout [41] with rate 0.5 before feeding this vector in the final tempo classification layer with a *softmax* function. During inference, quadratic interpolation of the output probability distribution is used to determine the final tempo in BPM.

The whole system has only $29{,}901$ trainable parameters, from which the multi-task tempo classification extension accounts for $5{,}100$. We contrast our compact model with the reported $2.9M$ parameters of the current state of the art in tempo estimation [40].

2.3 Network Training

To train the system, we represent annotated beat training data as impulse trains at the same temporal resolution as our input feature (i.e. 100 frames per second). To allow for slight deviations of the annotated beat locations and partially address the imbalance between the number of beat and non-beat frames, we use the neighbouring frames of the annotated beat positions as positive examples, but weight them by a scaling factor of 0.5 (cf. Figure 1d).

Given beat annotations, we derive tempo annotations by counting the inter-beat-intervals (IBI) to build a histogram. We smooth this histogram with a Hamming window of size 15 frames (i.e. 150 ms) to counteract small fluctuations of the beat annotations and determine the most dominant IBI by quadratic interpolation. This interval is then converted to tempo in BPM and mapped to tempo targets representing integer BPM values. In a similar way to the widening of the beat annotations, we smooth the tempo targets, but

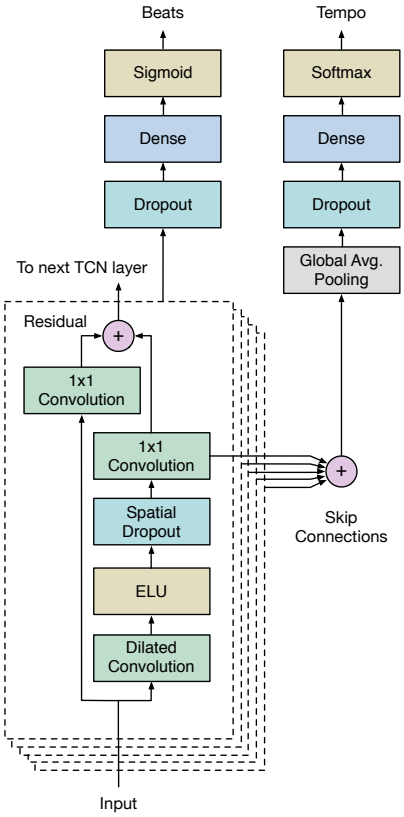


Figure 2: Structure of the neural network with the TCN for beat tracking (left) and the multi-task extension for tempo estimation (right).

extend the range to ± 2 BPM, weighting the neighbouring BPM targets with 0.5 and 0.25, respectively. We then normalise the tempo targets to form a probability distribution (as shown in Figure 1e) in order for it to be usable with the *softmax* activation function.

For training, we combine the cross-entropy losses of both network outputs by weighting them equally. Since the training sequences have different lengths, we train on whole sequences and minimise the combined loss with stochastic gradient descent (i.e. using a batch size of 1). We use *Adam* [28] with an initial learn rate of 0.002, and reduce it by a factor of 5 whenever the validation loss reaches a plateau and stop training if no improvement in validation loss is observed for 50 consecutive epochs or if a maximum of 150 epochs have elapsed. To avoid exploding gradients, we clip the gradients to a maximum norm of 0.5. If only tempo targets are present for training, we mask the loss of the beat tracking output. This way, only the error of the tempo output is backpropagated through the network and used to update the weights. It is important to note, that even in this scenario the shared beat and tempo feature representation gets adapted and optimised.

3. EXPERIMENTS AND EVALUATION

For experiments and evaluation we use the datasets listed in Table 1. Those listed in the upper part are used for training using 8-fold cross validation, and those in the lower part are independent test sets held back for evaluation only. If available, updated annotations are used and indicated by additional references. We chose these datasets in order to be able to compare the performance of our proposed systems to the best performing reference systems for both beat tracking and tempo estimation.

Dataset	files	length
Ballroom [23, 32] [1]	685	5 h 57 m
Beatles [12]	180	8 h 09 m
Hainsworth [24]	222	3 h 19 m
Simac [20]	595	3 h 18 m
SMC [27]	217	2 h 25 m
HJDB [25] $*$	235	3 h 19 m
ACM Mirum [35] $\star$	1410	15 h 05 m
GiantSteps [30, 39] $\star$	664	22 h 05 m
GTZAN [33, 43]	999	8 h 20 m

Table 1: Datasets used for training (upper half), and testing (lower half). The $*$ symbol denotes that only tempo annotations were used during training and beat annotations are used for evaluation only, and the $\star$ symbol indicates those datasets for which only tempo annotations exist.

The *HJDB* (Hardcore, Jungle, Drum & Bass) dataset is used to demonstrate the effectiveness of our multi-task extension w.r.t. its ability to improve beat tracking performance using only the tempo annotations of this set. This dataset was chosen, since its distinct music style is not well represented within any of the other training sets.

3.1 Beat Tracking Evaluation

We compare our proposed multi-task system to existing state-of-the-art beat tracking systems, namely to the underlying TCN approach presented in [11], and the two BLSTM approaches for beat [4] and joint beat and downbeat tracking [6]. Our goal is that the inclusion of the tempo classification layer is never detrimental to the performance of the beat tracking component.

Following the *de facto* standard for beat tracking evaluation, we report a set of different metrics with the parameterisation defined in [12]. We use the standard *F-measure*, as well as the continuity based measures *CMLc* and *CMLt* which require the beats to be tracked at the correct metrical level, as well as *AMLc* and *AMLt* which also allow alternate metrical interpretations such as double/half and offbeat. They either consider only the longest consecutive correctly tracked segment (*xMLc*) or all correctly tracked beats of a musical piece (*xMLt*).

From the results given in Table 2 it can be seen that all systems achieve essentially the same level of beat tracking accuracy, independent of the evaluation method. There are, however, smaller deviations from this general tendency. The beat output of the downbeat tracking system [6] performs slightly better on the *Ballroom* set, which might be

[1] The 13 identified duplicates were removed: `http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html`

due to the characteristic rhythmic patterns which can be better exploited by explicit modelling of whole bars.

	F	*CMLc*	*CMLt*	*AMLc*	*AMLt*
			Ballroom		
BLSTM [4]	0.917	0.832	0.849	0.905	0.926
BLSTM [6]	0.938	0.872	0.892	0.932	0.953
TCN [11]	0.933	0.864	0.881	0.909	0.929
Multi-task	0.931	0.864	0.883	0.908	0.930
			Hainsworth		
BLSTM [4]	0.884	0.769	0.808	0.873	0.916
BLSTM [6]	0.871	0.732	0.784	0.849	0.910
TCN [11]	0.874	0.755	0.795	0.882	0.930
Multi-task	0.877	0.756	0.798	0.880	0.928
			SMC		
BLSTM [4]	0.529	0.296	0.428	0.383	0.567
BLSTM [6]	0.516	0.307	0.406	0.429	0.575
TCN [11]	0.543	0.315	0.432	0.462	0.632
Multi-task	0.535	0.295	0.415	0.440	0.613
			GTZAN		
BLSTM [4]	0.864	0.750	0.768	0.901	0.927
BLSTM [6]	0.856	0.716	0.744	0.876	0.919
TCN [11]	0.843	0.695	0.715	0.889	0.914
Multi-task	0.847	0.702	0.724	0.886	0.916

Table 2: Beat tracking results on datasets used for training with 8-fold cross validation (top), and on completely unseen test data (bottom).

Given these results, we infer that the multi-task system achieves at least the same performance as the same system without the multi-task extension.

3.2 Multi-Task Evaluation

In the previous section, our evaluation focused on the use of both tempo- and beat-annotated training data within our multi-task model. In order to test our hypothesis that tempo-only information can indeed lead to improved beat tracking accuracy, and thus demonstrate the ability of multi-task learning to strengthen one target by learning additionally from the other, we perform a further experiment. To this end, we add a new dataset, but only use its tempo annotations for training.

We believe that the effect of this learning strategy should be most visible when performed with data, which is otherwise underrepresented in the training set. In our opinion, the *HJDB* dataset is a perfect fit since it contains musical genres from the early 1990s, namely Hardcore, Jungle, and Drum & Bass, which are characterised by their very distinct rhythmic structure. For the details on this dataset, see [25].

We train our new multi-task approach in two different ways. Once with the data as outlined in Table 1, but without *HJDB* (i.e. as in the previous section), and once including the tempo annotations of this dataset.

Inspection of the first two rows of Table 3 reveals that both the original TCN beat tracking system, and the system with the multi-task extension achieve roughly the same performance across all evaluation methods. However, once the additional tempo information is utilised (last row marked with the $*$ symbol), the performance increases by up to ~ 5 percentage points. The jump in accuracy is best observed in the *CMLc* and *CMLt* evaluation methods. This indicates that the system is able to exploit the additional information to track the beats at the correct metrical level more often than without this information. Within the context of the *HJDB* dataset where the "correct" metrical level is largely unambiguous, we consider this to be an important contribution.

	F	*CMLc*	*CMLt*	*AMLc*	*AMLt*
			HJDB		
TCN [11]	0.842	0.802	0.810	0.903	0.912
Multi-task	0.850	0.800	0.804	0.921	0.927
Multi-task $*$	0.882	0.848	0.858	0.937	0.947

Table 3: Multi-task learning beat tracking results on the *HJDB* dataset. All results obtained with 8-fold cross validation. The $*$ symbol denotes that tempo annotations of the *HJDB* set were used as additional targets during training.

3.3 Tempo Evaluation

Further to the beat tracking oriented evaluation results reported in the previous two sections, we also explore the effectiveness of our proposed approach for the task of global tempo estimation. To discover how our multi-task approach compares to the state of the art, we contrast its performance against four reference systems [5, 17, 36, 40]. Following the established evaluation practice for tempo estimation [23] we report the *Accuracy 1* and *Accuracy 2* scores with a tolerance of $\pm 4\%$ for each of these methods, with the results shown in Table 4.

Given that human perception of tempo is known to be subjective [34], this very reasonably manifests in multiple, valid interpretations of the beat among listeners and thus more than one acceptable tempo. Thus, in the context of automatic tempo estimation, it may not be realistic to expect to obtain near perfect performance on the *Accuracy 1* score on datasets of arbitrary musical makeup. To this end, we rely on the *Accuracy 2* score (which permits so-called "tempo octave errors") to better gauge performance.

On all of the reported datasets in Table 4, our proposed approach is the only one to consistently obtain an *Accuracy 2* greater than or equal to 0.938, which shows the high potential of our method to accurately find tempo across diverse musical data. Even with the stricter *Accuracy 1* evaluation, our method achieves at least a score of 0.697 which is ahead of all other methods, albeit by a small margin. It is important to stress that the *ACM Mirum*, *GiantSteps*, and *GTZAN* datasets are completely unseen by our multi-task approach, and this pattern even holds for *HJDB* when not included in the training set.

Concerning the *HJDB* set, we can observe a different overall pattern of performance compared to the other datasets, with a much smaller gap between *Accuracy 1* and

	Accuracy 1	*Accuracy 2*
ACM Mirum		
Gkiokas et al. [17]	0.725	0.979
Percival and Tzanetakis [36]	0.733	0.972
Böck et al. [5]	0.741	0.976
Schreiber and Müller [40]	0.795	0.974
Multi-task	0.757	0.977
Multi-task ∗	0.749	0.974
GiantSteps		
Gkiokas et al. [17]	0.721	0.922
Percival and Tzanetakis [36]	0.506	0.956
Böck et al. [5]	0.589	0.864
Schreiber and Müller [40]	0.730	0.893
Multi-task	0.697	0.958
Multi-task ∗	0.764	0.958
GTZAN		
Gkiokas et al. [17]	0.651	0.931
Percival and Tzanetakis [36]	0.658	0.924
Böck et al. [5]	0.697	0.950
Schreiber and Müller [40]	0.694	0.926
Multi-task	0.697	0.939
Multi-task ∗	0.673	0.938
HJDB		
Gkiokas et al. [17]	0.783	0.911
Percival and Tzanetakis [36]	0.285	1.0
Böck et al. [5]	0.796	0.868
Schreiber and Müller [40]	0.902	0.991
Multi-task	0.826	0.962
Multi-task ∗ †	1.0	1.0

Table 4: Tempo estimation results on completely unseen data. The ∗ symbol denotes that tempo annotations of the *HJDB* set were used as additional targets during training, the † symbol results obtained with 8-fold cross validation.

Accuracy 2 for most systems. Echoing the situation in the beat tracking evaluation in Table 3, we believe that this is a direct result of the unambiguous tempo for these styles of music. Looking across the performance of the other algorithms on *HJDB*, we discover that the method of Percival and Tzanetakis [36], while it also obtains a perfect score for *Accuracy 2*, is largely unable to identify the annotated tempo as shown by the disproportionately low score for *Accuracy 1*.

When trained with the additional tempo annotations of the *HJDB* set, our multi-task method is the only one able to detect the correct tempo for all pieces of this dataset for both *Accuracy 1* and then trivially for *Accuracy 2*. Although results are obtained with cross-validation, this was to be expected because of the homogeneity of the dataset. *Accuracy 1* on the *GiantSteps* set also greatly benefits from this additional training material, since this dataset contains a huge proportion of music labelled with the musical genre "drum and bass". On the other hand, having access to this kind of data (the system of Schreiber and Müller [40] was trained on an extended version of the *GiantSteps* dataset) can in turn result in very good scores on the *HJDB* set.

4. DISCUSSION AND CONCLUSIONS

In this paper, we have proposed a novel formulation for the simultaneous estimation of tempo and beat from musical audio signals within a multi-task learning framework. Via an extensive evaluation of both beat tracking and tempo estimation, we have demonstrated that our proposed multi-task approach leads to state-of-the-art performance across a wide variety of test datasets and relevant evaluation methods. Perhaps most critically, we have shown that, within this multi-task learning framework, we can improve the performance of beat tracking by providing it tempo-only annotations. In light of the challenges of obtaining high-quality annotated data for training beat tracking systems, the ability to profit from alternative training data which is both far more prevalent and easier to annotate, may have a significant impact on beat tracking moving forward.

In order to train our model, we made use of all of the available beat and tempo annotations within the allocated training sets in Table 1, and subsequently provided additional tempo-only annotations for evaluation on the *HJDB* dataset. We consider this split between beat and tempo annotated data to be one that is worthy of further exploration, in particular by seeking to understand how little beat annotated data is sufficient to achieve the same performance, assuming we can supplement the model with additional tempo annotations. This reduction of beat information could be posed in two ways, either by a lower number of fully annotated excerpts/pieces, or by restricting the duration of annotated sections across many pieces. If successful, the latter option would offer the possibility to rapidly increase the availability of training data by drastically reducing the burden of annotating long pieces of music—at least for those with roughly constant tempo.

We frame this discussion within the computational context of our proposed multi-task approach and the TCN beat tracker [11] which it extends. As previously stated, our multi-task model is highly effective in terms of objective performance, but with a fraction of the number of weights of other state-of-the-art approaches. This has two particularly beneficial properties. First, it allows for very efficient training (thanks in part to the ease of parallelisation of dilated convolutional models compared to recurrent architectures). Second, the training of networks with very few weights drastically reduces the degrees of freedom of the network, and hence strongly mitigates over-fitting. Thus, when looking beyond the limited domain of existing annotated datasets and considering generalisation capabilities of beat tracking and tempo estimation methods (and the subsequent re-use of this information for end-users) on totally unseen data, we believe that such "compact" deep models are worthy candidates for future research.

Supplementary material can be found online at `https://github.com/superbock/ISMIR2019` with executable code and pre-trained models being included in *madmom* [3] (`https://github.com/CPJKU/madmom`).

5. ACKNOWLEDGEMENTS

Sebastian Böck is supported by the Austrian Promotion Agency (FFG) under the "BASIS, Basisprogramm" umbrella program. Matthew E.P. Davies is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the project IF/01566/2015.

6. REFERENCES

[1] M. Alonso, G. Richard, and B. David. Accurate tempo estimation based on harmonic + noise decomposition. *EURASIP Journal on Applied Signal Processing*, pages 161–161, 2007.

[2] S. Bai, J. Z. Kolter, and V. Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018.

[3] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In *Proc. of the 2016 ACM Multimedia Conf.*, pages 1174–1178, 2016.

[4] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proc. of the 15th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 603–608, 2014.

[5] S. Böck, F. Krebs, and G. Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 625–631, 2015.

[6] S. Böck, F. Krebs, and G. Widmer. Joint beat and downbeat tracking with recurrent neural networks. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 255–261, 2016.

[7] S. Böck and M. Schedl. Enhanced beat tracking with context-aware neural networks. In *Proc. of the 14th Intl. Conf. on Digital Audio Effects (DAFx)*, pages 135–139, 2011.

[8] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997.

[9] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.

[10] D. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (ELUs). In *Proc. of the 4th Intl. Conf. on Learning Representations (ICLR)*, 2016.

[11] M. E. P. Davies and S. Böck. Temporal convolutional networks for musical audio beat tracking. In *Proc. of the 27th European Signal Processing Conf. (EUSIPCO)*, 2019.

[12] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, 2009.

[13] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.

[14] S. Dixon. An empirical comparison of tempo trackers. In *Proc. of the 8th Brazilian Symp. on Computer Music*, pages 832–840, 2001.

[15] A. Elowsson. Beat tracking with a cepstroid invariant neural network. In *Proc. of the 17th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 351–357, 2016.

[16] A. Gkiokas, V. Katsouros, and G. Carayannis. Reducing tempo octave errors by periodicity vector coding and SVM learning. In *Proc. of the 13th Intl Society for Music Information Retrieval Conf. (ISMIR)*, pages 301–306, 2012.

[17] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *Proc. of the 37th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–424, 2012.

[18] M. Goto and Y. Muraoka. A beat tracking system for acoustic signals of music. In *Proc. of the 2nd ACM Intl. Conf. on Multimedia*, pages 365–372, 1994.

[19] M. Goto. AIST annotation for the RWC music database. In *Proc. of the 7th Intl. Conf. on Music Information Retrieval (ISMIR)*, pages 359–360, 2006.

[20] F. Gouyon. *A computational approach to rhythm description — Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. PhD thesis, Universitat Pompeu Fabra, 2005.

[21] F. Gouyon and S. Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 25(1):34–54, 2005.

[22] F. Gouyon and P. Herrera. Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors. In *Audio Engineering Society Convention 114*, 2003.

[23] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.

[24] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 15:2385–2395, 2004.

[25] J. Hockman, M. E. P. Davies, and I. Fujinaga. One in the Jungle: Downbeat detection in Hardcore, Jungle, and Drum and Bass. In *Proc. of the 13th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 169–174, 2012.

[26] J. Hockman and I. Fujinaga. Fast vs slow: Learning tempo octaves from user data. In *Proc. of the 11th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 231–236, 2010.

[27] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *Proc. of the 3rd Intl. Conf. for Learning Representations (ICLR)*, 2015.

[29] A. Klapuri, A. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions Speech and Audio Processing*, 14(1):342–355, 2006.

[30] P. Knees, A. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. Le Goff. Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 364–370, 2015.

[31] F. Krebs, S. Böck, and G. Widmer. An efficient state space model for joint tempo and meter tracking. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 72–78, 2015.

[32] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 227–232, 2013.

[33] U. Marchand and G. Peeters. Swing ratio estimation. In *Proc. of the 18th Intl. Conf. on Digital Audio Effects (DAFx)*, pages 423–428, 2015.

[34] D. Moelants and M. McKinney. Tempo perception and musical content: What makes a piece fast, slow or temporally ambiguous. In *Proc. of the 8th Intl. Conf. on Music Perception and Cognition*, pages 558–562, 2004.

[35] G. Peeters and J. Flocon-Cholet. Perceptual tempo estimation using GMM-regression. In *Proc. of the 2nd ACM workshop on music information retrieval with user-centered and multimodal strategies (MIRUM)*, pages 45–50, 2012.

[36] G. Percival and G. Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1765–1776, 2014.

[37] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.

[38] H. Schreiber and M. Müller. A post-processing procedure for improving music tempo estimates using supervised learning. In *Proc. of the 18th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 235–242, 2017.

[39] H. Schreiber and M. Müller. A crowdsourced experiment for tempo estimation of electronic dance music. In *Proc. of the 19th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 409–415, 2018.

[40] H. Schreiber and M. Müller. A single-step approach to musical tempo estimation using a convolutional neural network. In *Proc. of the 19th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 100–105, 2018.

[41] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[42] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 648–656, 2015.

[43] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

[44] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu. Wavenet: A generative model for raw audio. *CoRR abs/1609.03499*, 2016.

[45] F.-H. F. Wu and J.-S. R. Jang. A supervised learning method for tempo estimation of musical audio. In *22nd Mediterranean Conf. of Control and Automation (MED)*, pages 599–604, 2014.

THE HARMONIX SET: BEATS, DOWNBEATS, AND FUNCTIONAL SEGMENT ANNOTATIONS OF WESTERN POPULAR MUSIC

Oriol Nieto[1] **Matthew McCallum**[1] **Matthew E. P. Davies**[2]
Andrew Robertson[3] **Adam Stark**[4] **Eran Egozy**[5]
[1] Pandora Media, Inc., Oakland, CA, USA
[2] INESC TEC, Porto, Portugal
[3] Ableton AG, Berlin, Germany
[4] MI·MU, London, UK
[5] MIT, Cambridge, MA, USA
onieto@pandora.com

ABSTRACT

We introduce the Harmonix set: a collection of annotations of beats, downbeats, and functional segmentation for over 900 full tracks that covers a wide range of western popular music. Given the variety of annotated music information types in this set, and how strongly these three types of data are typically intertwined, we seek to foster research that focuses on multiple retrieval tasks at once. The dataset includes additional metadata such as MusicBrainz identifiers to support the linking of the dataset to third-party information or audio data when available. We describe the methodology employed in acquiring this set, including the annotation process and song selection. In addition, an initial data exploration of the annotations and actual dataset content is conducted. Finally, we provide a series of baselines of the Harmonix set with reference beat-trackers, downbeat estimation, and structural segmentation algorithms.

1. INTRODUCTION

The tasks of beat detection [8], downbeat estimation [2], and structural segmentation [34] constitute a fundamental part of the field of MIR. These three musical characteristics are often related: downbeats define the first beat of a given music measure, and long structural music segments tend to begin and end on specific beat locations – frequently on downbeats [10]. The automatic estimation of such information could result in better musical systems such as more accurate automatic DJ-ing, better intra- and inter-song navigation, further musicological insights of large collections, *etc*. While a few approaches exploiting more than one of these musical traits have been proposed [2, 11, 25], the amount of human annotated data containing the three of them for a single collection is scarce. This limits the training potential of certain methods, especially those that require large amounts of information (e.g., deep learning [18]).

In this paper we present the Harmonix set: human annotations of beats, downbeats, and functional segmentation for 912 tracks of western popular music. These annotations were gathered with the aim of having a significant amount of data to train models to improve the prediction of such musical attributes, which would later be applied to various products offered by Harmonix, a video game company that specializes in musically-inspired games. By releasing this set to the public, our aim is to let the research community explore and exploit these annotations to advance the tasks of beat tracking, downbeat estimation, and automatic functional structural segmentation. We discuss the methodology to acquire these data, including the song selection process, and the inclusion of standard identifiers (AcoustID and MusicBrainz) and a set of automatically extracted onset times for the first 30 seconds of the tracks to allow other researchers to more easily access and align, when needed, the actual audio content. Furthermore, we present a series of results with reference algorithmic approaches in the literature with the goal of having an initial public benchmark of this set.

The rest of this work is organized as follows: Section 2 contains a review of the most relevant public datasets of the tasks at hand; Section 3 discusses the Harmonix set, including the data gathering, their formatting, and various statistics; Section 4 presents numerous benchmarks in the set; and Section 5 draws some final conclusions and discusses future work.

2. BACKGROUND

Several datasets with beat, downbeat, and/or segment annotations have been previously published, and in this section we review the most relevant ones.

2.1 Beat and Downbeat Tracking Sets

Over the last 15 years, many annotated datasets for beat and downbeat tracking have appeared in the literature whose primary purpose has been to allow the comparison of newly proposed and existing algorithms. However, the well-known difficulties of sharing the audio component of large annotated datasets has led to a rather ad-hoc usage of different datasets within the literature, and to a lesser extent, the choice of which evaluation metrics are selected to report accuracy. Conversely, the MIREX evaluation campaign provides a more rigid model for evaluation, by withholding access to private test datasets, and instead relying on the submission of the competing algorithms in order to compare them under controlled conditions. To this end, MIREX can be a useful reference point to consider these two music analysis tasks from the perspective of annotated data.

The MIREX Audio Beat Tracking (ABT) task [1] first appeared in 2006 and ran on a single dataset [28,30] with the performance of the submitted algorithms determined using one evaluation metric, the P-Score. After a brief hiatus, the task reappeared in 2009 with the addition of a dataset of Chopin Mazurkas [36], and the inclusion of multiple evaluation metrics [5]. The task continued to run in this way until the incorporation of the SMC dataset [16] in 2012, from which point it has remained constant. In 2014, the Audio Downbeat Estimation (ADE) task [2] was launched which comprised six different datasets from diverse geographic and stylistic sources: The Beatles [24]; Hardcore, Jungle, Drum and Bass (HJDB) [15]; Turkish [41]; Ballroom [21]; Carnatic [42]; and Cretan [17], with the evaluation conducted using the F-measure. While the datasets contained with these two MIREX tasks are by no means exhaustive, they provide a useful window to explore both how the audio data is chosen and how the annotation is conducted for these MIR tasks. To this end, we provide the following breakdown of different properties including reference to both MIREX and non-MIREX datasets.

Duration: Unlike the task of structural segmentation, beat and downbeat tracking datasets can be comprised of musical excerpts [14, 15, 21, 28] rather than full tracks [9,12,13,24]. **Number of annotators**: The initial MIREX beat tracking dataset [28] was unique in that it contained the annotations of 40 different people who tapped the beat to the music excepts. Conversely, other datasets used multiple annotators contributing across the dataset [16], a single annotator for all excerpts [14], or even deriving the annotations in a semi-automatic way from the output of an algorithm [24]. **Annotation post-processing**: Given some raw tap times or algorithm output, these can either be left unaltered [28] or, as is more common, iteratively adjusted until they are considered perceptually accurate by the annotator(s) [14–16]. **Style-specificity**: While some datasets are designed to have broad coverage across a range of musical styles [13, 14, 23], others target a particular group of styles [15, 21], a single style [9], the work of a given artist [12, 24] or even multiple versions of the same pieces [36]. **Western / Non-Western**: Similarly, the make up of the dataset can target underrepresented non-western music [33,41,42]. **Perceived difficulty**: Finally, the choice of musical material can be based upon the perceived difficulty of the musical excerpts, either from the perspective of musical or signal level properties [16].

2.2 Structural Segmentation Sets

The task of structural segmentation has been particularly active in the MIR community since the late 2000s. Similarly to the beat tracking task, several datasets have been published, and some of them have evolved over time. This task is often divided into two subtasks: segment boundary retrieval and segment labeling. All well-known published datasets contain both boundary and label information. One of the major challenges with structural segmentation is that this task is regarded as both *ambiguous* (i.e., there may be more than one valid annotation for a given track [26]) and *subjective* (i.e., two different listeners might perceive different sets of segment boundaries [4]). This has led to different methodologies when annotating and gathering structural datasets, thus having a diverse ecosystem of sets to choose from when evaluating automatic approaches.

The first time this task appeared on MIREX was in 2009, [3] where annotations from The Beatles dataset (which also includes beat and downbeat annotations, as previously described) and a subset of the Real World Computing Popular Music Database (RWC) [13] were employed. These sets contain several functional segment annotations for western (The Beatles) and Japanese (RWC) popular music. These segment functions describe the *purpose* of the segments, e.g.: "solo," "verse," "chorus." A single annotation per track is provided for these two sets. The Beatles dataset was further revised at the Tampere University of Technology, [4] and additional functional segment annotations for other bands were added to The Beatles set, which became known as the Isophonics Dataset [24]. No beat or downbeat annotations were provided to the rest of the tracks in Isophonics, and the final number of tracks with functional structural segment annotations is 300. The number of annotated tracks in RWC is 365.

To address the open problems of ambiguity and subjectivity, further annotations per track from several experts could be gathered. That is the case with the Structural Annotations for Large Amounts of Music Information (SALAMI) dataset [39], where most of its nearly 1,400 tracks have been annotated by at least 2 musical experts. Similarly, the Structural Poly Annotations of Music (SPAM) dataset [32] provides 5 different annotations for 50 tracks. These two sets not only contain functional levels of annotations, but also large and small scale segments where only single letters describing the similarity between segments are annotated. Thus, these can be seen as sets that contain *hierarchical* data, which pose significant chal-

[1] https://www.music-ir.org/mirex/wiki/2006:Audio_Beat_Tracking
[2] https://www.music-ir.org/mirex/wiki/2014:Audio_Downbeat_Estimation
[3] https://www.music-ir.org/mirex/wiki/2009:Structural_Segmentation
[4] http://www.cs.tut.fi/sgn/arg/paulus/beatles_sections_TUT.zip

lenges, since ambiguity and subjectivity span across multiple layers [26] and remain largely unexploited in the MIREX competition [7,40]. As opposed to Isophonics and RWC, these two sets contain highly diverse music in terms of genre: from world-music to rock, including jazz, blues, and live music.

The following properties typically define segmentation datasets: **Number of annotators**: This can help when trying to quantify the amount of disagreement among annotators [26, 32], or when developing approaches that may yield more than one potentially valid segmentation. **Hierarchy**: The levels of annotations contained in the set. It typically contains functional, large, and/or small segment annotations. When only one level of annotations is provided, these are typically called *flat* segment annotations.

3. THE HARMONIX SET

In this section we present the Harmonix set, including the methodology of acquiring the data, its motivation, its contents, and a set of annotation statistics. The Harmonix set is publicly available on-line. [5]

3.1 Data Gathering

The primary motivation of this work is based on the need to create gameplay data for rhythm-action games (also known as beat matching games). Many such games exist, from early pioneers like Parappa The Rapper and Beatmania, to the rock simulation games Guitar Hero and Rock Band, as well as community-based games like OSU and more recently, VR games like Beat Saber. In most cases the gameplay data (also referred to as beatmaps), consisting of note locations in a song, are hand-authored. In certain games, additional control data may be desirable. For example, in the rock simulation games, where a 3D depiction of a rock concert is rendered, it can be desirable to simulate flashing lights (on the beat) or lighting color palette changes (on section boundaries). Again, these data tend to be hand-authored.

Harmonix's desire was to implement a suite of automatic music analysis tools that estimate certain musical attributes in order to expedite the process of hand-authoring gameplay data, or in some cases, to fully automate the process of creating these data. The songs of the Harmonix set were gathered and hand-annotated to create a ground-truth dataset for training and testing these algorithms.

The mix of genres in this corpus were chosen to be typical of ones used in the rhythm-action games, with a somewhat higher tendency towards EDM and popular songs suitable for dancing (see Figure 1 for the full genre distribution). As such, most tend to have a very stable tempo and a 4/4 time signature. However, we also added a selection of songs that may not be typical of dance or pop music to increase variety. Some of these (Classic Rock, Country, Metal) may have less stable tempo (where drums are played by actual musicians as opposed to drum-machines

[5] https://github.com/urinieto/harmonixset

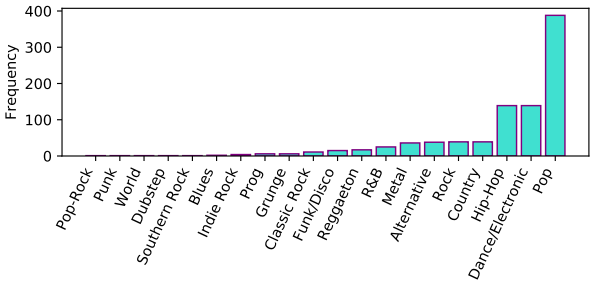


Figure 1. Genre distribution of the Harmonix set.

or DAW-based productions) and may deviate from a strict 4/4 meter.

All songs were annotated by trained professional musicians who regularly work in music production environments. As the project went on, the majority of annotation work fell to only a few individuals who became specialized in this task. Annotations were created in Digital Audio Workstation software (such as Reaper or Logic). First, a MIDI tempo track was established that corresponded to the song audio. Then beats, downbeats, and sections were coded into the MIDI track using note events and text events. MIDI files were then exported and automatically converted to a text-based representation of beats, downbeats, and named section boundaries. Every song was verified once by the original annotator.

3.2 Dataset Contents

The Harmonix set contains manual annotations for 912 western popular music tracks, thus being the largest published dataset to date containing beats, downbeats, and function structural segmentation information. The annotations and some of the song-level metadata are distributed via JAMS [19] files, one per track. This format is chosen given its simplicity when storing multi-task annotations plus song- and corpus-level metadata. Each JAMS file contains the beat, downbeat, and functional segmentation annotations, plus a set of estimated onsets for the first 30 seconds of the audio. These onsets are intended to help aligning the audio in case researchers obtain audio data with different compression formats that might include certain small temporal offsets. This onset information was computed using librosa [27], with their default parameters. [6]

For the sake of transparency and usability, we also publish the raw beats, downbeats, and segmentation data as space-separated text files, two per track: one for beats and downbeats, and the other for segments. We also distribute the code that converts these raw annotations into unified JAMS files. Furthermore, we provide other identifiers with the aim of easily retrieving additional metadata and/or audio content for each song. These identifiers include:

- **MusicBrainz** [7] : open music encyclopedia including

[6] librosa 0.6.3, using Core Audio on macOS 10.13.6.
[7] https://musicbrainz.org/

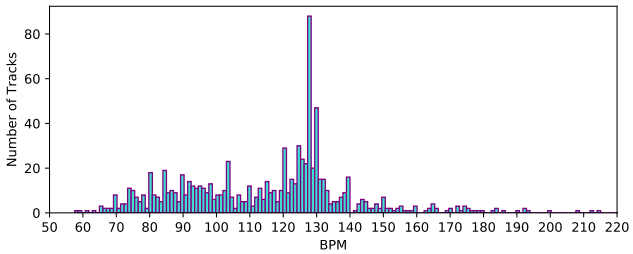


Figure 2. Tempo distribution of the tracks in the set.

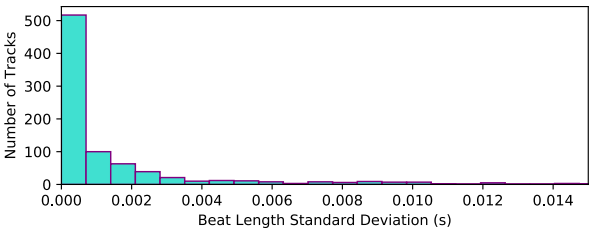


Figure 3. Standard deviation of the tempo distribution.

unique identifiers for recordings, releases, artists, etc.

- **AcoustID** [8]: open source fingerprinting service to easily match audio content, typically associated with MusicBrainz identifiers.

Finally, we provide a single CSV file including additional metadata information such as genre, time signature, and BPM.

3.3 Data Statistics

In this subsection we provide several data insights obtained from the annotations to give an objective overview of the set. In Figure 2 we show the estimated tempo distribution in beats-per-minute (BPM) per track. These estimations were computed using the track-level median inter-beat-interval (IBI) for each of the annotated beats in a given track. There is a clear peak at 128 BPM, which could be explained by being the most common tempo in electronic dance music [29]. Furthermore, in Figure 3 we plot the standard deviation of the IBI. We can clearly see that the tempo is remarkably steady in this dataset, which is expected given the type of musical genres it spans.

In terms of segment statistics, we show data based on certain attributes described in a MIREX meta-analysis of the segmentation task [40]. In Figure 4 we plot track-level histograms for the number of segments, and the number of unique segments (i.e., those with the same associated label). Both distributions seem to be unimodal and centered around 10 and 11 for the number of segments per tracks, and around 6 and 7 for the number of unique labels per track. This differs from the number of unique segments in The Beatles dataset, which is centered around 4 per track [31].

[8] https://acoustid.org/

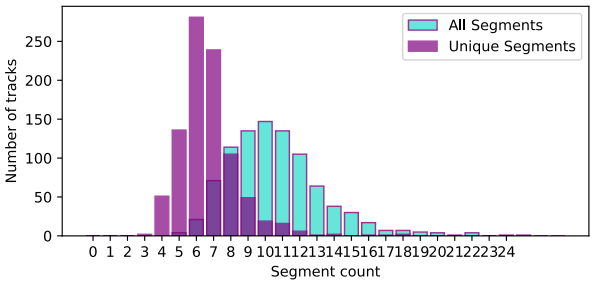


Figure 4. Number of segments per track, based on their segment labels.

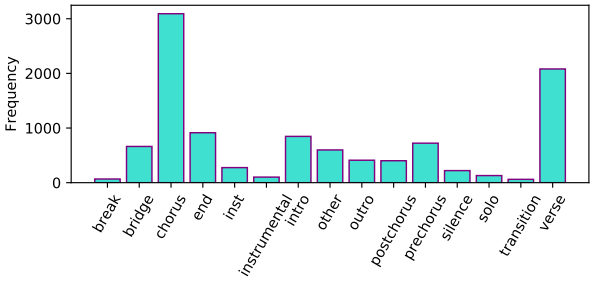


Figure 5. Most common segment labels.

Figure 5 shows the frequency in which the most common segment labels appear in the set. The labels "chorus" and "verse" dominate the distribution, as these functional parts are common in western popular music. The plot also shows potentially repeated labels like "inst" and "instrumental." A further inter-song analysis of the labels might be necessary to potentially merge certain labels and thus unify the vocabulary of the set.

We plot in Figure 6 the distribution of the segment lengths, in seconds, across the entire dataset. As we showed in Figure 2, there is a majority of tracks at 128 BPM, for which a duration of 15 seconds would correspond to a segment of exactly 32 beats. This, in the common 4/4 time signature, would result in 8 bars per each 15-second segment in that tempo, and 8 bars are common in electronic dance music [29].

Finally, and thanks to having access to the annotated downbeats, we show in Figure 7 the number of segments

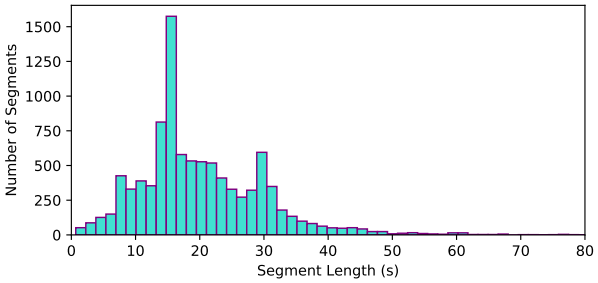


Figure 6. Segment length distribution.

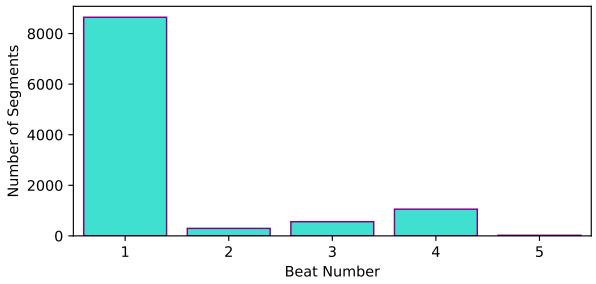


Figure 7. Number of segments based on their starting beat position within a bar.

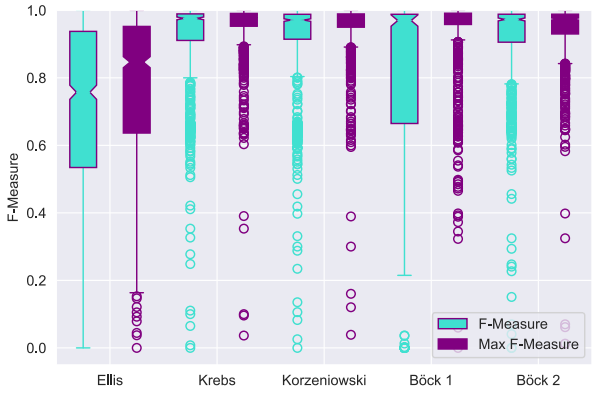


Figure 8. Beat tracking performance over the Harmonix set, for the algorithms Ellis [8], Krebs [22], Korzeniowski [20], Böck 1 - the "BeatDetector" technique from [3], and Böck 2 - the "BeatTracker" technique from [3].

starting at a specific beat within a given bar. We can see that the vast majority of segments (81.1%) start in a downbeat. Interestingly, several segments (10%) start in position 4, thus showing that 1-beat count-ins are more common than other types of count-ins on this dataset (a popular example of a 1-beat count-in song is Hey Jude by The Beatles, where the (1) is on the Jude and Hey is the (4) of the previous bar).

4. RESULTS

4.1 Beat Results

In order to establish performance baselines over the dataset for the task of beat-tracking, we have evaluated a number of openly available beat tracking algorithms on the dataset [3,8,20,22]. Each of these algorithms can be found in either the madmom [1] or librosa python libraries. [9] By running these algorithms in other datasets with the same metrics, a comparison of datasets could ultimately be performed. The results are also included in the dataset repository in CSV format. This is intended as a convenience for any future work that wishes to evaluate novel algorithms against these benchmarks.

The beat tracking results for the aforementioned algorithms are displayed in Figure 8. They are evaluated across two metrics, F-Measure, and Max F-Measure, where the latter refers to the maximum F-Measure obtained per track when evaluated across double and half-time metrical variations in the annotated beats provided with this dataset. In all experiments a tolerance window of ± 70 ms was employed in order to compute the F-Measure. For half-time metrical variations, both the downbeat and upbeat alignments were tested for a maximum F-Measure value. While [8] is the most computationally efficient of the algorithms, we see clear gains in the more recently developed methods. When investigating the types of errors present in the beat position estimates from [8], it was found the most common error was the alignment of beat phase. Often beat positions landed on the half beat or quarter beat, resulting in an F-Measure of 0 when this misalignment is consistent throughout the track. When comparing F-Measure and Max F-Measure metrics, it can be seen that with this dataset both [8] and the "BeatDetector" algorithm from [3] have a significant number of double-half time errors, compared to the other algorithms evaluated. Unlike the "BeatTracker" algorithm in [3], the "BeatDetector" algorithm assumes constant tempo.

4.2 Downbeat Results

Unfortunately, the availability of open source downbeat estimation libraries is limited. In order to provide a baseline for downbeat detection performance with the Harmonix set specifically, results have been evaluated with the downbeat detection algorithms available in [1] in addition to Durand's algorithm [6] [10], making three algorithms in total. The algorithms from the madmom python package [1] include the method proposed in [2] using the annotated beat positions as input, and the dynamic Bayesian bar tracking processor using the input from the RNN bar processor activation function. The results can be seen in Figure 9 in terms of F-Measure with a tolerance window of ± 70 ms. The superior performance of [2], which has oracle annotated beat information, highlights the importance of reliable beat tracking for downbeat estimation performance, and the interdependence between the beat tracking and downbeat estimation tasks.

4.3 Segmentation Results

There are several open source structural segmentation algorithms available in the Music Structure Analysis Framework (MSAF) [32]. [11] We run the best performing ones on the Harmonix set: (i) Structural Features [38] to identify boundaries, and (ii) 2D-Fourier Magnitude Coefficients (2D-FMC) [31] to label the segments based on their acoustic similarity. Constant-Q Transforms [37] are the selected

[9] We used madmom 0.16.1 and librosa 0.6.3. We noticed a bias in this librosa version where beats were offset by a consistent number of milliseconds. More specifically, we employed librosa's `beat.beat_track` method with default arguments on macOS 10.13.6.

[10] Not open source, shared via private correspondence.

[11] MSAF version dev-0.1.8.

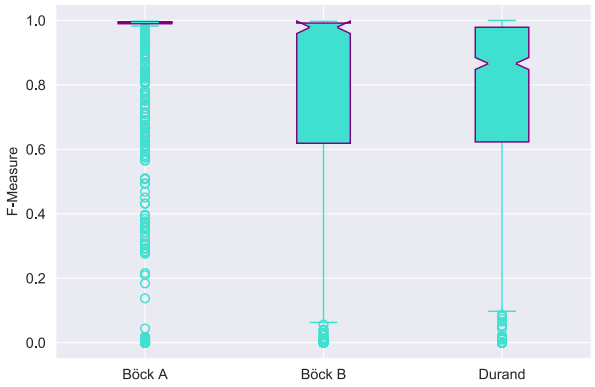


Figure 9. Downbeat tracking performance over the Harmonix set, for the algorithms Böck A [2] and Böck B - a dynamic Bayesian network provided within the madmom package [1], and Durand [6].

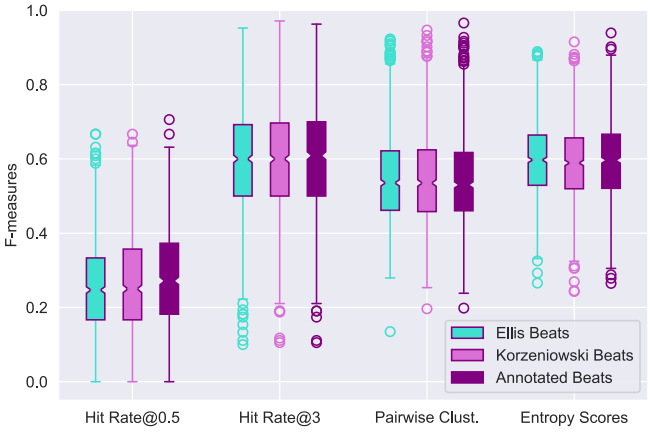


Figure 10. Segmentation results over the Harmonix set, using Structural Features for boundaries, 2D-FMC for the labeling process, and three types of beat information.

audio features given their ability to capture both timbral and harmonic content, and the default parameters in MSAF are the ones employed when computing these results. We use `mir_eval` [35] to evaluate these algorithms, and report the F-measures for the most common metrics: Hit Rate with 0.5 and 3 second windows for boundary retrieval, and Pairwise Frame Clustering and Entropy Scores for the labeling process. These algorithms can use beat-synchronized features, and we ran each algorithm three times, depending on the following beat information: (i) Ellis' estimations, (ii) Korzeniowski's estimations, and (iii) annotations from the Harmonix set. Thus, we are able to assess the segmentation results when employing the worst and best performing beat trackers from our previous study, plus those computed using human annotated beats. Song-level results for these three different runs are available as CSV files in the dataset repository disclosed above.

In Figure 10 all segmentation results are shown. The results in turquoise boxplots (on the left side) display the metrics of the algorithms when running on Ellis' beat-synchronized features, those in light pink (in the middle) correspond to the results computed with Korzeniowski's beats, while the purple boxplots (on the right) show those using annotated beats instead. Given how related boundary retrieval is with respect to precise beat placement, it is not unexpected to see an improvement in the boundary metrics (Hit Rates) when using more accurate beat data. The boxplots further show that the smaller the time window used in the Hit Rate metrics the more accurate the beat information should ideally be. In other words, Korzeniowki's beats yield very similar results than those from human annotations when using a 3 second window, but there is clearly room for enhancement (in terms of beat tracking) when using 0.5 second windows, where the segmentation results using human annotated beats outperform any of the others that employ estimated ones. On the other hand, it is worth noting that the label results do not seem to depend as much on the quality of the beats in order to produce their outcomes, as the three different runs yield similar results for the Pairwise Frame Clustering and Entropy Scores metrics. As mentioned in Section 2.2, structural segmentation is a challenging task especially due to ambiguity, subjectivity, and hierarchy, and this is reflected in the overall results, which exhibit notable room for improvement.

5. CONCLUSIONS

We presented the Harmonix set, the largest dataset in terms of human annotations containing the following three types of music information: beats, downbeats, and function structural segments. This set contains mostly western popular music, with strong emphasis on Pop, EDM, and Hip-Hop. We provide metadata in terms of genre, song title, and artist information along with standard identifiers such as MusicBrainz and AcoustID plus predicted onset information to allow easier matching and alignment with audio data. We discussed a set of results using current algorithms in the literature in terms of beat tracking, downbeat estimation, and structural segmentation to disclose an initial public benchmark of the set. Given the rather large nature of the set and the three different types of music information contained in it, it is our hope that researchers employ these data not only to further advance one of these three MIR tasks individually, but also to potentially combine them to yield superior approaches in the near future.

6. ACKNOWLEDGMENTS

We would like to thank Simon Durand for sharing his downbeat estimation implementation. Matthew E.P. Davies is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the project IF/01566/2015.

7. REFERENCES

[1] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In *Proceedings of the 24th*

ACM international conference on Multimedia, pages 1174–1178, 2016.

[2] S. Böck, F. Krebs, and G. Widmer. Joint Beat and Downbeat tracking with recurrent neural networks. *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference*, pages 255–261, 2016.

[3] S. Böck and M. Schedl. Enhanced beat tracking with context-aware neural networks. In *Proc. Int. Conf. Digital Audio Effects*, pages 135–139, 2011.

[4] M. J. Bruderer, M. F. McKinney, and A. Kohlrausch. The Perception of Structural Boundaries in Melody Lines of Western Popular Music. *Musicæ Scientiæ*, 13(2):273–313, 2009.

[5] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, 2009.

[6] S. Durand, J. P. Bello, B. David, and G. Richard. Feature adapted convolutional neural networks for downbeat tracking. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 296–300. IEEE, 2016.

[7] A. F. Ehmann, M. Bay, J. S. Downie, I. Fujinaga, and D. D. Roure. Music Structure Segmentation Algorithm Evaluation: Expanding on MIREX 2010 Analyses and Datasets. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 561–566, Miami, FL, USA, 2011.

[8] D. P. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.

[9] V. Eremenko, E. Demirel, B. Bozkurt, and X. Serra. Audio-aligned jazz harmony dataset for automatic chord transcription and corpus-based research. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 483–490, 2018.

[10] J. T. Foote. Methods for the automatic analysis of music and audio. *FXPAL Technical Report FXPAL-TR-99-038*, 1999.

[11] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello. A Music Structure Informed Downbeat Tracking System Using Skip-Chain Conditional Random Fields and Deep Learning. In *Proc. of the 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019.

[12] B. D. Giorgi, M. Zanoni, S. Böck, and A. Sarti. Multipath beat tracking. *Journal of the Audio Engineering Society*, 64(7/8):493–502, 2016.

[13] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases. *International Conference on Music Information Retrieval*, (October):287–288, 2002.

[14] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 15:2385–2395, 2004.

[15] J. Hockman, M. E. P. Davies, and I. Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *Proc. of the 13th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 169–174, 2012.

[16] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.

[17] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the "odd": Meter inference in a culturally diverse music corpus. In *Proc. of the 15th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 425–430, 2014.

[18] E. J. Humphrey, J. P. Bello, and Y. LeCun. Moving Beyond Feature Design: Deep Architecture and Automatic Feature Learning in Music Informatics. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 403–408, Porto, Portugal, 2012.

[19] E. J. Humphrey, J. Salamon, O. Nieto, J. Forsyth, R. M. Bittner, and J. P. Bello. JAMS: A JSON Annotated Music Specification for Reproducible MIR Research. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 591–596, Taipei, Taiwan, 2014.

[20] F. Korzeniowski, S. Böck, and G. Widmer. Probabilistic extraction of beat positions from a beat activation function. In *ISMIR*, pages 513–518, 2014.

[21] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 227–232, 2013.

[22] F. Krebs, S. Böck, and G. Widmer. An efficient state-space model for joint tempo and meter tracking. In *ISMIR*, pages 72–78, 2015.

[23] U. Marchand and G. Peeters. Swing ratio estimation. In *Proc. of the 18th Intl. Conf. on Digital Audio Effects (DAFx)*, pages 423–428, 2015.

[24] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. OMRAS2 Metadata Project 2009. In *Late Breaking Session of the 10th International Society of Music Information Retrieval*, Kobe, Japan, 2009.

[25] M. C. McCallum. Unsupervised Learning of Deep Features for Music Segmentation. In *Proc. of the 44th International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brighton, UK, 2019.

[26] B. McFee, O. Nieto, M. M. Farbood, and J. P. Bello. Evaluating hierarchical structure in music annotations. *Frontiers in Psychology*, 8(1337), 2017.

[27] B. McFee, C. Raffel, D. Liang, D. P. W. Ellis, M. McVicar, E. Battenberg, and O. Nieto. librosa: Audio and Music Signal Analysis in Python. In *Proc. of the 14th Python in Science Conference*, pages 18–25, Austin, TX, USA, 2015.

[28] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri. Evaluation of audio beat tracking and music tempo extraction algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.

[29] D. Moelants. Hype vs. Natural Tempo: a Long-term Study of Dance Music Tempi. In *Proc. of the 10th International Conference on Music Perception and Cognition*, Sapporo, Japan, 2008.

[30] D. Moelants and M. McKinney. Tempo perception and musical content: What makes a piece fast, slow or temporally ambiguous. In *Proc. of the 8th Intl. Conf. on Music Perception and Cognition*, pages 558–562, 2004.

[31] O. Nieto and J. P. Bello. Music Segment Similarity Using 2D-Fourier Magnitude Coefficients. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 664–668, Florence, Italy, 2014.

[32] O. Nieto and J. P. Bello. Systematic Exploration of Computational Music Structure Research. In *Proc. of the 17th International Society for Music Information Retrieval Conference*, pages 547–553, New York City, NY, USA, 2016.

[33] L. O. Nunes, M. Rocamora, L. Jure, and L. W. Biscainho. Beat and Downbeat Tracking Based on Rhythmic Patterns Applied to the Uruguayan Candombe Drumming. In *Proc. of the 16th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, pages 264–270, 2015.

[34] J. Paulus, M. Müller, and A. Klapuri. Audio-Based Music Structure Analysis. In *Proc of the 11th International Society of Music Information Retrieval*, pages 625–636, Utrecht, Netherlands, 2010.

[35] C. Raffel, B. Mcfee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir_eval: A Transparent Implementation of Common MIR Metrics. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 367–372, Taipei, Taiwan, 2014.

[36] C. Sapp. Comparative Analysis of Multiple Musical Performances. In *Proc. of the 8th Intl. Conf. on Music Information Retrieval, (ISMIR)*, pages 497–500, 2007.

[37] C. Schörkhuber and A. Klapuri. Constant-Q Transform Toolbox for Music Processing. In *Proc. of the 7th Sound and Music Computing Conference*, pages 56–64, Barcelona, Spain, 2010.

[38] J. Serrà, M. Müller, P. Grosche, and J. L. Arcos. Unsupervised Music Structure Annotation by Time Series Structure Features and Segment Similarity. *IEEE Transactions on Multimedia, Special Issue on Music Data Mining*, 16(5):1229 – 1240, 2014.

[39] J. B. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie. Design and Creation of a Large-Scale Database of Structural Annotations. In *Proc. of the 12th International Society of Music Information Retrieval*, pages 555–560, Miami, FL, USA, 2011.

[40] J. B. L. Smith and E. Chew. A meta-analysis of the MIREX Structure Segmentation task. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 251–256, Curitiba, Brazil, 2013.

[41] A. Srinivasamurthy, A. Holzapfel, and X. Serra. In search of automatic rhythm analysis methods for turkish and indian art music. *Journal of New Music Research*, 43(1):94–114, 2014.

[42] A. Srinivasamurthy and X. Serra. A Supervised Approach to Hierarchical Metrical Cycle Tracking from Audio Music Recordings. In *Proc. of the 39th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5237–5241, 2014.

DECONSTRUCT, ANALYSE, RECONSTRUCT: HOW TO IMPROVE TEMPO, BEAT, AND DOWNBEAT ESTIMATION

Sebastian Böck
enliteAI, Vienna, Austria
s.boeck@enlite.ai

Matthew E. P. Davies
University of Coimbra, CISUC, DEI
mepdavies@dei.uc.pt

ABSTRACT

In this paper, we undertake a critical assessment of a state-of-the-art deep neural network approach for computational rhythm analysis. Our methodology is to deconstruct this approach, analyse its constituent parts, and then reconstruct it. To this end, we devise a novel multi-task approach for the simultaneous estimation of tempo, beat, and downbeat. In particular, we seek to embed more explicit musical knowledge into the design decisions in building the network. We additionally reflect this outlook when training the network, and include a simple data augmentation strategy to increase the network's exposure to a wider range of tempi, and hence beat and downbeat information. Via an in-depth comparative evaluation, we present state-of-the-art results over all three tasks, with performance increases of up to 6% points over existing systems.

1. INTRODUCTION

A central concept in much of the work on audio beat tracking is the "tactus" – described as the most comfortable foot-tapping rate when unconsciously tapping to a piece of music. As stipulated by London [1, Ch.1] (and references therein), the tactus is essential for our perception of metre. The tactus by itself carries no information concerning the metrical organisation within a piece of music, but it is informative about both local and global tempo. To perceive metre, we require the hierarchical organisation between at least two levels, and ideally three: a level above the tactus which indicates the longer-term grouping of beats into bars (or measures), and a lower level to describe how the beats are sub-divided – whether in *simple* time (divided by two), or *compound* time (divided by three).

In this sense, we can expand the notion of (unmarked) foot-tapping towards "counting" in time to music. While numerous counting systems exist for the teaching of musical rhythm [2], the "traditional" American system is perhaps the most well-known. For two-level counting, we can mark the three beats of the bar of a waltz as follows: **1** 2 3 **1** 2 3 **1**..., where the **1** indicates the first beat of each bar, the downbeat. Moving to three-level counting, we can count the sub-divisions of a four beat bar into two as: **1** + 2 + 3 + 4 + **1**..., (*one - and - two - and - three - and - four - and*), and the sub-divisions of the same four beat bar into four as: **1** e + a 2 e + a 3 e + a 4 e + a **1**... (*one - "ee" - and - "ah" - two - "ee" - and -"ah"* and so on).

From the perspective of computational rhythm analysis, we can thus make a distinction between approaches which target one metrical level in isolation, as opposed to those which estimate more than one. Among the single-level approaches, the vast most majority fall within the domain of beat-tracking (e.g [3–6]). When the focus of the analysis moves towards downbeats, this almost exclusively relies on the implicit or explicit modelling of another metrical level, either the beat [7], tatum [8], or a contrast between both [9]. One notable outlier is the downbeat prediction approach of Jehan [10] which relies instead on onset-synchronous analysis.

Concerning the modelling of three simultaneous metrical levels, few published approaches exist. Goto [11] presents a real-time system for estimating the quarter-note, half-note, and measure levels, but doesn't address the sub-beat level. Klapuri et al. [12] on the other hand, address the estimation of tatum, beat, and downbeat, with explicit dependencies between the phase of the beat and the tatum, and the period of the beat and downbeat. For a recent review of beat and downbeat estimation, see [13].

Considering the topic of tempo estimation, which, in most instances, seeks to retrieve a single value to describe a global tempo, existing approaches can be split into two categories: those which make their estimate of the tempo based the post-processing of a sequence of beat times (e.g. for a discussion of techniques, see [14]) and those which treat the task as a classification or regression problem and do not require any prior estimate of the beats [15–18].

In this paper, we seek to work from the perspective of leveraging shared connections in musical structure, and address the simultaneous estimation of three highly interconnected properties of musical rhythm: tempo, beat, and downbeat within a single model. In line with much of the recent literature concerning the extraction of musical information from audio signals [19], we adopt a deep learning approach. We depart from our recent multi-task approach [20] for tempo and beat estimation using a temporal convolutional network (TCN), which was shown to provide state-of-the-art results. We undertake a critical assessment of its constituent parts, and on the basis of our

analysis, adapt it in several ways. At the broadest level, we wish to leverage the benefit of modelling a metrical hierarchy (as opposed to just the beat level) by the inclusion of an additional learning task, downbeat estimation. In terms of the structure of the network itself, we adapt the shallowest layers of the network (i.e. those closest to the musical surface) to provide a better model of harmonic musical sounds. In addition, we propose a novel formulation of the TCN architecture which incorporates an additional dilation rate to each layer as a means to embed understanding of integer ratios modelling the metrical structure.

A peculiar aspect of the evaluation in [20] was the ability of the multi-task model to perfectly estimate the tempo of the HJDB dataset [21] when it was included in the training splits, with good, but noticeably lower performance when it was left as a hold-out test set. Given the characteristic fast tempo of *HJDB*, we speculate that the gap in performance arose due to the lack of any similarly fast-tempo music in the training sets. Following this argument, a secondary motivation of this work is to consider how data augmentation can be used in an efficient way to extrapolate information from regions of the training data which are well-covered in terms of tempo annotations to those which are more sparse.

Via a thorough evaluation across the three tasks of tempo, beat, and downbeat estimation, we demonstrate state-of-the-art performance, and draw attention to the ability of TCN-based approaches to leverage shared representations for multi-task analysis of musical audio signals.

The remainder of this paper is structured as follows. In Section 2 we describe our multi-task formulation and data augmentation strategy in detail. In Section 3 we present an ablation study and comparative evaluation against existing reference systems. Finally, in Section 4 we discuss the impact of the contribution and promising areas for future work.

2. APPROACH

Our earlier multi-task approach for tempo and beat estimation [20] was itself an extension of an earlier TCN-based approach for beat tracking [22]. The core component, which is common to both, is a deep neural network (DNN) architecture based on dilated convolutions, most well-known from *WaveNet* [23]. It is quite striking to consider that an architecture designed for the causal generation of raw audio (primarily for speech synthesis), and with its roots in an auto-regressive process, can find application in a problem cast as binary classification through time, i.e. the classification per frame of the presence or absence of a beat. From an alternative perspective, we may view the strength of the TCN in this problem domain as resulting from multiple connections (both forwards and backwards in time) at different time scales, and thus bearing similarity to much earlier work on the cognitively-inspired use of multi-resolution signal processing for beat tracking [24].

For a detailed description of the existing architecture, we refer to the reader to [20, 22]. In brief, the multi-task approach uses a log-magnitude spectrogram with 81 logarithmically spaced frequency bins and a frame rate of 100 frames per second as input. Overlapping spectrogram snippets are passed through 3 convolution and max pooling layers, followed by 11 dilated convolutional layers whose dilation rate increases by a factor of 2 per layer. The so-called "skip connections" between these layers are provided as an auxiliary output of the TCN and are used to generate a prediction of the tempo across a linear range from $0 - 300$ beats per minute (bpm). The main output of the TCN, a beat activation function, is then processed by a dynamic Bayesian network (DBN) [25] to obtain a final sequence of beat estimates.

In spite of the reported high performance of the multi-task approach on a wide range of musical material for both beat and tempo estimation, we believe it is valuable to question the design decisions of this network and consider the ways in which it could be modified to improve performance. Our focus in this paper is on the core of the network, namely the convolutional and max pooling layers together with the TCN. In a coarse sense, we can consider the convolutional and max pooling layers to relate to more surface-level properties of the music and hence local information, i.e. *what are the spectro-temporal properties of the beats?* with the deeper TCN layers oriented more towards their temporal dependency over longer time scales, i.e. *how is the beat and metrical structure organised over the duration of musical pieces?*

Concerning the first question, a common limitation of beat tracking systems is their ability to reliably detect the beat in music without the presence of drums, as typified, at least in part, by lower reported performance in classical music. Given the high prevalence of rock, pop, jazz, and electronic dance music among existing beat tracking datasets [26], we consider the modelling of harmonic sounds to be important when addressing under-represented musical styles and of crucial importance to reliably detect downbeats in Western music, where harmonic changes often occur at bar boundaries [27]. Regarding the second question, we directly enable the network to learn feature representations which are integer multiplies of each other by deploying multiple concurrent dilated convolutions at each TCN layer, and in so doing embed some implicit hierarchical structure into the model.

In what follows, we describe the specific modifications made to the network. Since the work in this paper explicitly targets the improvement of an existing approach, we allude to performance increases wherever relevant, with detailed results in Section 3.

2.1 Multi-task formulation

Based on the model described above, we add the additional task of downbeat tracking. This can be accomplished in various ways. One option is to model the downbeats and beats jointly as a multi-class problem, i.e. by classifying each input frame to be a beat, a downbeat, or neither. This approach was successfully deployed in [28], but has the downside that it cannot fully leverage the information if a dataset contains only beat or downbeat annotations. Thus

we treat the problem as a multi-label classification problem instead, with the downbeat task treated as a separate binary classification problem with its own output. We model the downbeat output similarly to the beat output as a single *sigmoid* unit which is fed directly from the main TCN output. Whereas the approach in [20] used 16 filters per layer for the multi-task estimation of tempo and beat, with the addition of the downbeat task, we expand the network to include more filters and increase this to 20.

This additional output is then also post-processed with a DBN. Since the beat and downbeat outputs do not define a joint probability density function (i.e. their sum is not guaranteed to be 1 as for multi-class problems), the DBN post-processing used in [28] cannot be applied directly to the combined beat and downbeat activations. Thus the difference of the beat and downbeat activations (limited to positive values) and the downbeat activations are used as state-conditional observations for beats and downbeats, respectively. In Section 3 we refer to this approach as *joint downbeat tracking*.

An alternative approach is to first detect the beats and then in a second inference step to find the downbeats given the set of beat predictions. This approach was chosen in [29] and has the most notable advantage that the large joint state space which is required to model multiple bar lengths and tempi at a frame level resolution can be split into two smaller ones. The first one (tracking the beats) only requires multiple tempi to be modelled at the frame level resolution, whereas the second one operates at beat resolution and is completely tempo invariant, thus requiring only very few states. The downside of this approach is that errors made in beat tracking directly propagate to downbeat tracking. In Section 3 we refer to this approach as *sequential downbeat tracking*.

2.2 Conv layers

Both the original beat tracking paper [22] and the multi-task extension tackling global tempo estimation presented in [20] use the same convolutional block to reduce multiple consecutive STFT frames to a one-dimensional feature vector which is then processed by the TCN. Two groups of alternating 3×3 convolution and 1×3 max-pooling layers were used to reduce overlapping spectrogram windows of size 5×81 (time $\times$ frequency) down to 3×26 and 1×8 before these eight bands (roughly representing one octave each) were combined into a singular value with a 1×8 convolution. This feature representation is closely related to the one used in [12] and was shown to work well.

However, a musically motivated reordering of these layers can have a positive effect on the performance of the model. Convolutional filters covering multiple frequency bins but only a single time step have been shown to concentrate on harmonic and timbral features [30] and proven to work well for multiple tasks, including key estimation [31] and automatic music transcription [32]. Moving the "frequency only" convolution in between the two 3×3 convolutions as shown in Figure 1, enables the network to better capture harmonic content across a wider frequency range instead of detecting local changes in smaller regions of the spectrogram only and then later aggregating them.

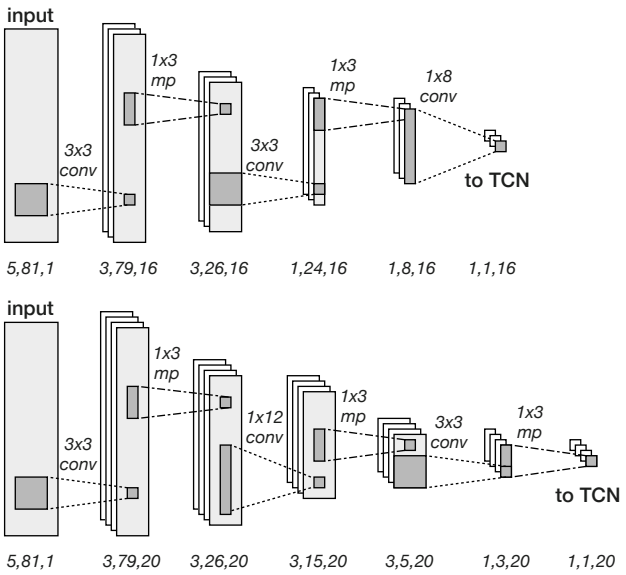


Figure 1: Comparison of the convolution (*conv.*) and max pooling (*mp*) layers. The architecture from [20, 22] (top). Our proposed architecture (bottom). The dimensions of the tensors are shown below each layer.

2.3 TCN layers

From a musical perspective, it is undeniable that discovering downbeats requires more knowledge about the signal than locating beat positions only. Independently of whether this additional knowledge is harmonic or rhythmic in nature, it always requires a longer temporal context. Increasing the temporal context of the TCN by either using larger kernel sizes or adding more layers (with exponentially increasing dilation rates), did not improve any of the tasks under investigation. This observation is not necessarily surprising since the temporal context modelled by the TCN is already about 40 seconds – which should be sufficient to tackle the task of tracking the locations of the downbeats and estimating the length of the bars. Instead, adding a second dilated convolution (with a doubled dilation rate) to each of the TCN layers enables the network to simultaneously model musical properties at various levels which are integer multiples of each other. We discovered that adding a third dilation rate did not further improve performance, but we believe this is very likely an artefact of the data utilised for training, since none of the datasets used have a noticeable number of musical pieces with compound time signatures. The feature maps of the two dilated convolutions are concatenated before spatial dropout [33] and an exponential linear unit (ELU) activation function [34] is applied. In order to keep the output dimensionality of the TCN layer constant, these feature maps are then combined by a 1×1 convolution, which increases the total number of parameters linearly with each TCN layer instead of exponentially.

2.4 Data augmentation

Our approach to data augmentation is both simple and straightforward and similar to the scaling approach applied in [17]. Contrary to other data augmentation strategies, which pre-process the audio signal and manipulate it in various ways (e.g. time stretching, pitch shifting, sample rate conversion to simulate speeding up or slowing down the signal [35, 36]), we do not change the audio signal itself, but instead only change the parameters of the STFT when obtaining a time-frequency representation. To be more precise, we only change the rate at which the overlapping frames of the STFT are obtained from the audio signal by sampling from a normal distribution with 5% standard deviation from the annotated tempo. By changing only the hop size, we obtain spectrograms with varying overlap factors and only the targets have to be adjusted accordingly. Using this data augmentation strategy leads to many more training examples for tempi which are otherwise underrepresented in the data, as can be seen in Figure 2.

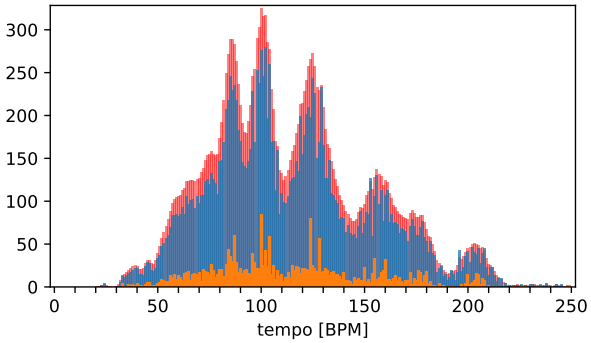


Figure 2: Tempo distribution of original tempo annotations (orange, foreground), after data augmentation (blue) and target widening (red, background).

2.5 Network training

Using data augmentation increases the amount of data the network can learn from. However, this also leads to increased training times when using conventional training procedures. Furthermore, the additional downbeat classification layer, the inclusion of a second dilated convolution and the usage of more filters in each of the TCN layers has a notable impact on the size of the model, which now has $116,302$ trainable parameters compared to $29,901$ of [20].

To this end, we make use of the latest training optimisation strategies, namely *RAdam* [37] and *Lookahead Optimization* [38]. The combination of these two drastically reduces the training time (even accounting for the larger number of weights) simultaneously leading to models being less sensitive to different random initialisations. All remaining hyper-parameters were left unchanged. We found the used learning rate of $2e^{-3}$ and clipping the gradients at a norm of 0.5 a sensible choice, as is training on full sequences with a batch size of 1.

We derive the tempo targets in the same way by computing a smoothed and interpolated histogram on the inter beat intervals. We apply the same target widening strategy to present the network not only the annotated frame and tempo, but also their direct neighbouring frames and ± 2 BPM values as positive targets, albeit with lower weights of 0.5 and 0.25, respectively.

3. EXPERIMENTS AND RESULTS

We use the same datasets as in [20] with the most recent annotations available. *Beatles* [39], *Cuidado* [40], *Hainsworth* [20, 41], *Simac* [42], *SMC* [26], and *HJDB* [21, 28] are used for training and evaluated in an 8-fold cross validation manner. *ACM Mirum* [43, 44], *GiantSteps* [45, 46], and *GTZAN* [47, 48] are used as test datasets. Predictions for the test datasets are obtained by averaging the predictions of the networks trained for cross validation. To enable future comparisons, we make all annotations as well as the beat, downbeat, and tempo estimates available at the accompanying website. [1]

For evaluation, we use the standard metrics used in the literature. For tempo estimation, we report *Accuracy 1* and *Accuracy 2* scores with a tolerance of $\pm 4\%$ as used in [49]. For beat and downbeat tracking, we use *F-measure* and the continuity based metrics *CMLt* (requires beats being tracked at the annotated metrical level) and *AMLt* (allowing alternative metrical levels, such as double/half and triple/third tempo as well as off-beat) with the tolerances as defined as in [39].

3.1 Ablation study

Before reporting comparative evaluation to other methods, we aim to understand how each of the proposed measures outlined in Section 2 contribute to the final performance of the system.

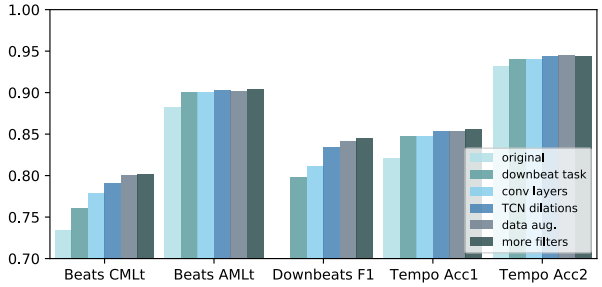


Figure 3: Impact of the improvements proposed for selected evaluation metrics. Mean values over the complete validation set are given.

From Figure 3 it can be seen that the measures undertaken to improve the original system do not contribute the same to the different tasks and the given evaluation metrics. For example, beat tracking *AMLt* and tempo *Accuracy 2* scores increase only marginally, which is best explained by the fact that the baseline system is already performing at a high level on these tasks. However, since these metrics allow metrical ambiguities, it is impossible

[1] https://github.com/superbock/ISMIR2020

to determine if the system is considering the correct metrical level in the case of beat tracking or the correct tempo octave. Both *CMLt* and *Accuracy 1* require the reported beat locations and tempo to exactly match the annotations (within the allowed tolerance). These metrics therefore better catch the ability of an algorithm to correctly predict the annotated information.

Concentrating on these metrics, it can be seen that additionally modelling and predicting downbeats has a positive effect on beat tracking and tempo estimation. This effect is then strengthened by the modifications made to the convolutional and TCN layers. Using data augmentation and more filters gives a small additional boost. It should be noted that the positive effect of data augmentation on the generalisation capabilities of the network are mostly visible for the task of tempo estimation if "out of tempo distribution" datasets are used for evaluation. Since the validation set is a randomly chosen subset of the training set (and hence has a very similar tempo distribution), the impact is not fully reflected in Figure 3.

3.2 Tempo estimation

Tempo estimation is the task with the most noticeable overall impact of the proposed refinements. While *Accuracy 2* values have been quite high for many systems among all datasets under consideration, the new system is the only one consistently achieving high *Accuracy 1* values as well (Table 1). The system's ability to model several tasks simultaneously and exploit mutual information relevant to all tasks leads to an increased performance of more than 6% points in *Accuracy 1* over the best results reported so far on certain datasets.

	Accuracy 1	*Accuracy 2*
ACM Mirum		
Gkiokas et al. [50]	0.725	0.979
Percival and Tzanetakis [44]	0.733	0.972
Schreiber and Müller [17]	0.781	0.976
Böck et al. [20]	0.749	0.974
Foroughmand & Peeters [18]	0.733	0.965
Ours	0.841	0.990
GiantSteps		
Gkiokas et al. [50]	0.721	0.922
Percival and Tzanetakis [44]	0.506	0.956
Schreiber and Müller [17] *	0.821	0.971
Böck et al. [20]	0.764	0.958
Foroughmand & Peeters [18] *	0.836	0.979
Ours	0.870	0.965
GTZAN		
Gkiokas et al. [50]	0.651	0.931
Percival and Tzanetakis [44]	0.658	0.924
Schreiber and Müller [17]	0.769	0.926
Böck et al. [20]	0.673	0.938
Foroughmand & Peeters [18]	0.697	0.891
Ours	0.830	0.950

Table 1: Tempo estimation results on unseen test data. Asterisks denote systems which have been trained on a disjoint set of the same source.

3.3 Beat tracking

Although beat tracking performance of existing systems is already very high, the new system sets new high scores in *CMLt* and even exceeds the very high performance values above 0.9 (on *Ballroom*) by more than 4% points. Other systems achieve such high scores only under the less strict *AMLt* metric, which also permits metrical errors, including double/half, triple/third tempo, and off-beat. This highlights the capability of the system to track beats exactly at the annotated metrical level.

	F-measure	*CMLt*	*AMLt*
Ballroom			
Böck et al. [28]	0.938	0.892	0.953
Elowsson [51] ‡	0.925	0.903	0.932
Davies and Böck [22]	0.933	0.881	0.929
Ours (beat tracking)	0.956	0.935	0.958
Ours (joint tracking)	0.962	0.947	0.961
Hainsworth			
Böck et al. [5]	0.884	0.808	0.916
Elowsson [51] ‡	0.742	0.676	0.792
Davies and Böck [22]	0.874	0.795	0.930
Ours (beat tracking)	0.904	0.851	0.937
Ours (joint tracking)	0.902	0.848	0.930
SMC			
Böck et al. [5]	0.529	0.428	0.567
Elowsson [51] ‡	0.375	0.225	0.332
Davies and Böck [22]	0.543	0.432	0.632
Ours (beat tracking)	0.552	0.465	0.643
Ours (joint tracking)	0.544	0.443	0.635
GTZAN			
Böck et al. [5]	0.864	0.768	0.927
Davies and Böck [22]	0.843	0.715	0.914
Ours (beat tracking)	0.883	0.808	0.930
Ours (joint tracking)	0.885	0.813	0.931

Table 2: Beat tracking results on datasets used for training with 8-fold cross validation (top), and on unseen test data (bottom). ‡ was trained on *Ballroom* data only.

In Table 2 it can also be seen that joint modelling of beats and downbeats (in the DBN) can be beneficial for music with constant meter and steady tempo (e.g. *Ballroom*), whereas it negatively impacts performance for expressive music as contained in *Hainsworth* and *SMC*.

3.4 Downbeat tracking

For the task of downbeat tracking the systems, performance can be clearly separated into two main categories: i) the systems of Durand et al. [8] and Fuentes et al. [9], which explicitly model harmonic features (using chroma features as input for the neural network) and ii) the ones of Böck et al. [28] and ours which learn harmonic features implicitly. Whereas the former show better performance on pop music (e.g. the *Beatles* dataset) where downbeats often coincide with harmonic changes, they perform less well on data where bars are mostly defined based on rhythm.

	F-measure	*CMLt*	*AMLt*
Ballroom			
Böck et al. [28]	0.863	0.834	0.931
Durand et al. [8]	0.797	0.616	0.916
Fuentes et al. [9]	0.83	-	-
Ours (sequential tracking)	0.900	0.894	0.953
Ours (joint tracking)	0.916	0.913	0.960
Hainsworth			
Böck et al. [28]	0.684	0.628	0.832
Durand et al. [8]	0.664	0.500	0.804
Fuentes et al. [9]	0.67	-	-
Ours (sequential tracking)	0.713	0.686	0.855
Ours (joint tracking)	0.722	0.696	0.872
Beatles			
Böck et al. [28]	0.831	0.730	0.858
Durand et al. [8]	0.847	0.722	0.875
Fuentes et al. [9]	0.86	-	-
Ours (sequential tracking)	0.829	0.748	0.860
Ours (joint tracking)	0.837	0.742	0.862
GTZAN			
Böck et al. [28]	0.640	0.577	0.824
Durand et al. [8]	0.607	0.480	0.774
Ours (sequential tracking)	0.654	0.619	0.817
Ours (joint tracking)	0.672	0.640	0.832

Table 3: Downbeat tracking results on datasets used for training with 8-fold cross validation (top), and on unseen test data (bottom).

Regarding the question of whether *joint downbeat tracking* or *sequential downbeat tracking* is superior, Table 3 shows a consistent advantage for processing beats and downbeats simultaneously. The only exception is the *Beatles* dataset, which contains some music with changing metre. Due to memory constraints, joint downbeat tracking cannot model these metre changes. Modelling them is computationally only feasible with sequential downbeat tracking, which may further benefit from sub-beat modelling, as used in [9].

4. DISCUSSION AND CONCLUSIONS

In this paper we address the multi-task estimation of three inter-related properties of musical metre: tempo, beat, and downbeat. Our approach is somewhat unconventional as we do not propose a new method from scratch, but instead we deconstruct, analyse, and then reconstruct an existing approach as a means to further the state of the art. By pairing our methodology with an ablation study, we are able to directly observe the impact of the implemented changes, and in turn, to observe the cumulative gains in performance. Via our evaluation, it is clear that there is no "magic bullet" among our proposed modifications, yet their combination is clearly effective. Furthermore, we must accept that when the baseline performance is already high, the margin for improvement is somewhat limited.

By close inspection of the performance of our approach in comparison both to the baseline and other existing systems, we consider the main impact of our approach as constituting a "closing of the gap" between stricter and more lenient evaluation metrics across each of the tasks. For tempo estimation, our approach is the first to exceed 0.83 for *Accuracy 1* across three large reference datasets, which are completely unseen to our training scheme. Likewise, when considering the positive impact for beat tracking, we find the clearest improvements in the evaluation metric which enforces tracking at the annotated metrical level. Since the relative improvements under the more lenient metrics are much smaller, we do not believe that our approach has unlocked the means to accurately infer the tempo, beat, or downbeat in extremely challenging musical examples. Reference to the incremental improvements for the *SMC* dataset for beat tracking can immediately attest to this. Indeed, the lack of improvement for this kind of musical material may require the reformulation of the inference techniques used to recover the final outputs, rather than intervention at the point of training the networks. Alternatively, they may require a fundamentally different way in which to present targets to the network which is better able to model temporal uncertainty in the annotations. We consider both of these to be promising areas for future work in order to address more challenging data in a robust way.

Ultimately, we believe the main contribution of our work rests in the increased reliability of the good predictions made by the model across these three tasks. It is well-established within music cognition that the perception of tempo, beat, and metre is ambiguous and varies among listeners; therefore within the MIR community, it is easy to justify the use of "multiple-choice" evaluation methodologies. However, this evaluation practice explicitly masks the fact that for almost any piece of music, at least some of these allowed options will be much less reasonable than others. Thus, in the absence of a multi-level annotation methodology in which the set of allowed annotations are specific to individual pieces of music, the only way to guarantee a high-quality prediction (in an unsupervised way) is to aim to maximise performance under stricter evaluation metrics. The alternative is to perform a subjective assessment of beat and downbeat performance via listening to clicks mixed with the audio signals. Given the large amount of musical material in existing datasets, this remains a daunting prospect. However, by restricting this kind of supervised analysis to the subset of excerpts which are accurate only when allowing for alternative interpretations of the annotations, we may move towards a closer estimate of the true performance of these systems. In addition, this kind of partial subjective evaluation could act as a means to "bootstrap" the specification of alternative hypotheses on a per-excerpt basis.

5. ACKNOWLEDGMENTS

This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Opera-

tional Program Centro 2020, as well as by Portuguese National Funds through the FCT - Foundation for Science and Technology, I.P., under the project IF/01566/2015.

We wish to thank Kazuyoshi Yoshii and Leigh M. Smith for inspiring discussions which helped shape this paper.

6. REFERENCES

[1] J. London, *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, 2004.

[2] S. L. Gage, "An analysis and comparison of rhythm instructional materials and techniques for beginning instrumental music students," Ph.D. dissertation, University of Illinois at Urbana-Champaign, 1994.

[3] E. D. Scheirer, "Tempo and beat analysis of acoustic musical signals," *The Journal of the Acoustical Society of America*, vol. 103, no. 1, pp. 588–601, 1998.

[4] B. McFee and D. P. W. Ellis, "Better beat tracking through robust onset aggregation," in *Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2014, pp. 2154–2158.

[5] S. Böck, F. Krebs, and G. Widmer, "A multi-model approach to beat tracking considering heterogeneous music styles." in *Proc. of the 15th Intl. Society for Music Information Retrieval Conf.*, 2014, pp. 603–608.

[6] T. Cheng, S. Fukayama, and M. Goto, "Convolving Gaussian Kernels for RNN-Based Beat Tracking," in *Proc. of the 26th European Signal Processing Conf.*, 2018, pp. 1919–1923.

[7] G. Peeters and H. Papadopoulos, "Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 6, pp. 1754–1769, 2011.

[8] S. Durand, J. P. Bello, B. David, and G. Richard, "Robust downbeat tracking using an ensemble of convolutional networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 76–89, 2017.

[9] M. Fuentes, B. McFee, H. C. Crayencour, S. Essid, and J. P. Bello, "Analysis of common design choices in deep learning systems for downbeat tracking," in *Proc. of the 19th Intl. Society for Music Information Retrieval Conf.*, 2018, pp. 106–112.

[10] T. Jehan, "Downbeat prediction by listening and learning," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2005, pp. 267–270.

[11] M. Goto, "An audio-based real-time beat tracking system for music with or without drum-sounds," *Journal of New Music Research*, vol. 30, no. 2, pp. 159–171, 2001.

[12] A. Klapuri, A. Eronen, and J. Astola, "Analysis of the meter of acoustic musical signals," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 1, pp. 342–355, 2006.

[13] M. Fuentes, "Multi-Scale Computational Rhythm Analysis: A Framework for Sections, Downbeats, Beats, and Microtiming," Ph.D. dissertation, Université Paris-Saclay, 2019.

[14] H. Schreiber, "Data-driven approaches for tempo and key estimation of music recordings," Ph.D. dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), 2020.

[15] A. J. Eronen and A. P. Klapuri, "Music tempo estimation with k-nn regression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 50–57, 2009.

[16] H. Schreiber and M. Müller, "A post-processing procedure for improving music tempo estimates using supervised learning." in *Proc. of the 18th Intl. Society for Music Information Retrieval Conf.*, 2017, pp. 235–242.

[17] ——, "A single-step approach to musical tempo estimation using a convolutional neural network," in *Proc. of the 19th Intl. Society for Music Information Retrieval Conf.*, 2018, pp. 100–105.

[18] H. Foroughmand and G. Peeters, "Deep-rhythm for global tempo estimation in music," in *Proc. of the 20th Intl. Society for Music Information Retrieval Conf.*, 2019, pp. 636–643.

[19] H. Purwins, B. Li, T. Virtanen, J. Schlüter, S.-Y. Chang, and T. Sainath, "Deep learning for audio signal processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 2, pp. 206–219, 2019.

[20] S. Böck, M. E. P. Davies, and P. Knees, "Multi-task learning of tempo and beat: Learning one to improve the other," in *Proc. of the 20th Intl. Society for Music Information Retrieval Conf.*, 2019, pp. 486–493.

[21] J. Hockman, M. E. P. Davies, and I. Fujinaga, "One in the Jungle: Downbeat detection in Hardcore, Jungle, and Drum and Bass," in *Proc. of the 13th Intl. Society for Music Information Retrieval Conf.*, 2012, pp. 169–174.

[22] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proc. of the 27th European Signal Processing Conf.*, 2019.

[23] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," *CoRR abs/1609.03499*, 2016.

[24] L. M. Smith, "A multiresolution time-frequency analysis and interpretation of musical rhythm," Ph.D. dissertation, University of Western Australia, 2000.

[25] F. Krebs, S. Böck, and G. Widmer, "An efficient state space model for joint tempo and meter tracking," in *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, 2015, pp. 72–78.

[26] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon, "Selective sampling for beat tracking evaluation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2539–2548, 2012.

[27] H. Papadopoulos and G. Peeters, "Joint estimation of chords and downbeats from an audio signal," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 138–152, 2010.

[28] S. Böck, F. Krebs, and G. Widmer, "Joint beat and downbeat tracking with recurrent neural networks," in *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.*, 2016, pp. 255–261.

[29] F. Krebs, S. Böck, M. Dorfer, and G. Widmer, "Downbeat tracking using beat-synchronous features and recurrent neural networks," in *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.*, 2016, pp. 129–135.

[30] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *Proc. of 14th Intl. Workshop on Content-Based Multimedia Indexing*, 2016.

[31] H. Schreiber and M. Müller, "Musical tempo and key estimation using convolutional neural networks with directional filters," in *Proc. of the Sound and Music Computing Conf.*, 2019, pp. 47–54.

[32] R. Kelz, S. Böck, and G. Widmer, "Deep polyphonic ADSR piano note transcription," in *Proc. of the IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2019, pp. 129–135.

[33] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler, "Efficient object localization using convolutional networks," in *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 648–656.

[34] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," in *Proc. of the 4th Intl. Conf. on Learning Representations*, 2016.

[35] B. McFee, E. Humphrey, and J. Bello, "A software framework for musical data augmentation," in *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, 2015, pp. 248 – 254.

[36] J. Schlüter and T. Grill, "Exploring data augmentation for improved singing voice detection with neural networks." in *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, 2015, pp. 121–126.

[37] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han, "On the variance of the adaptive learning rate and beyond," in *Proc. of the 8th Intl. Conf. on Learning Representations*, 2020.

[38] M. R. Zhang, J. Lucas, G. Hinton, and J. Ba, "Lookahead optimizer: k steps forward, 1 step back," in *Proc. of the 33rd Conf. on Neural Information Processing Systems*, 2019.

[39] M. E. P. Davies, N. Degara, and M. D. Plumbley, "Evaluation methods for musical audio beat tracking algorithms," Centre for Digital Music, Queen Mary University of London, Tech. Rep. C4DM-TR-09-06, 2009.

[40] F. Gouyon and P. Herrera, "Determination of the meter of musical audio signals: Seeking recurrences in beat segment descriptors," in *Audio Engineering Society Convention 114*, 2003.

[41] S. Hainsworth and M. Macleod, "Particle filtering applied to musical tempo tracking," *EURASIP Journal on Applied Signal Processing*, vol. 15, pp. 2385–2395, 2004.

[42] F. Gouyon, "A computational approach to rhythm description — audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing," Ph.D. dissertation, Universitat Pompeu Fabra, 2005.

[43] G. Peeters and J. Flocon-Cholet, "Perceptual tempo estimation using GMM-regression," in *Proc. of the 2nd ACM workshop on music information retrieval with user-centered and multimodal strategies (MIRUM)*, 2012, pp. 45–50.

[44] G. Percival and G. Tzanetakis, "Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1765–1776, 2014.

[45] P. Knees, A. Faraldo, P. Herrera, R. Vogl, S. Böck, F. Hörschläger, and M. Le Goff, "Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections." in *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, 2015, pp. 364–370.

[46] H. Schreiber and M. Müller, "A crowdsourced experiment for tempo estimation of electronic dance music," in *Proc. of the 19th Intl. Society for Music Information Retrieval Conf.*, 2018, pp. 409–415.

[47] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[48] U. Marchand and G. Peeters, "Swing ratio estimation," in *Proc. of the 18th Intl. Conf. on Digital Audio Effects*, 2015, pp. 423–428.

[49] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano, "An experimental comparison of audio tempo induction algorithms," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, p. 1832–1844, 2006.

[50] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis, "Music tempo estimation and beat tracking by applying source separation and metrical relations," in *Proc. of the 37th IEEE Intl. Conf. on Acoustics, Speech and Signal Processing*, 2012, pp. 421–424.

[51] A. Elowsson, "Beat tracking with a cepstroid invariant neural network," in *Proc. of the 17th Intl. Society for Music Information Retrieval Conf.*, 2016, pp. 351–357.

Tapping Along to the Difficult Ones: Leveraging User-Input for Beat Tracking in Highly Expressive Musical Content

António Sá Pinto[1,2(✉)] and Matthew E. P. Davies[1]

[1] INESC TEC, Sound and Music Computing Group, Porto, Portugal
{antonio.s.pinto,matthew.davies}@inesctec.pt
[2] Faculdade de Engenharia da Universidade do Porto, Porto, Portugal

Abstract. We explore the task of computational beat tracking for musical audio signals from the perspective of putting an end-user directly in the processing loop. Unlike existing "semi-automatic" approaches for beat tracking, where users may select from among several possible outputs to determine the one that best suits their aims, in our approach we examine how high-level user input could guide the manner in which the analysis is performed. More specifically, we focus on the perceptual difficulty of tapping the beat, which has previously been associated with the musical properties of expressive timing and slow tempo. Since musical examples with these properties have been shown to be poorly addressed even by state of the art approaches to beat tracking, we re-parameterise an existing deep learning based approach to enable it to more reliably track highly expressive music. In a small-scale listening experiment we highlight two principal trends: i) that users are able to consistently disambiguate musical examples which are easy to tap to and those which are not; and in turn ii) that users preferred the beat tracking output of an expressive-parameterised system to the default parameterisation for highly expressive musical excerpts.

Keywords: Beat tracking · Expressive timing · User input

1 Introduction and Motivation

While the task of computational beat tracking is relatively straightforward to define – its aim being to replicate the innate human ability to synchronise with a musical stimulus by tapping a foot along with the beat – it remains a complex and unsolved task within the music information retrieval (MIR) community. Scientific progress in MIR tasks is most often demonstrated through improved accuracy scores when compared with existing state of the art methods [25]. At the core of this comparison rest two fundamental tenets: the (annotated) data upon which the algorithms are evaluated, and the evaluation method(s) used to measure performance. In the case of beat tracking, both the tasks of annotating datasets of musical material and measuring performance are non-trivial [10]. By

R. Kronland-Martinet et al. (Eds.): CMMR 2019, LNCS 12631, pp. 75–90, 2021.
https://doi.org/10.1007/978-3-030-70210-6_5

its very nature, the concept of beat perception – how an individual perceives the beat in a piece of music – is highly subjective [21]. When tapping the beat, listeners may agree over the phase, but disagree over the tempo or preferred metrical level – with one tapping, *e.g.*, twice as fast as another, or alternatively, they may agree over the tempo, but tap in anti-phase. This inherent ambiguity led to the prevalence of multiple hypotheses of the beat, which can arise at the point of annotation, but more commonly appear during evaluation where different interpretations of ground truth annotations are obtained via interpolation or sub-sampling. In this way, a wide net can be cast in order not to punish beat tracking algorithms which fail to precisely match the annotated metrical level or phase of the beats; with this coming at the expense that some unlikely beat outputs may inadvertently be deemed accurate. Following this evaluation strategy, the performance of the state of the art is now in the order of 90% on existing datasets [4,5] comprised primarily of pop, rock and electronic dance music. However, performance on more challenging material [15] is considerably lower, with factors such as expressive timing (*i.e.*, the timing variability that characterises a human performance, in opposition to a metronomic or "perfectly" timed rendition [11]), recording quality, slow tempo and metre changes among several identified challenging properties.

Although beat tracking has garnered much attention in the MIR community, it is often treated as an element in a more complex processing pipeline which provides access to "musical time", or simply evaluated based on how well it can predict ground truth annotations. Yet, within the emerging domain of creative-MIR [16,22] the extraction of the beat can play a critical role in musically-responsive and interactive systems [18]. A fundamental difference of applying beat tracking in a creative application scenario is that there is a specific end-user who wishes to directly employ the music analysis and thus has very high expectations in terms of its performance [1]. To this end, obtaining high mean accuracy scores across some existing databases is of lower value than knowing "*Can the beats be accurately extracted (as I want them) for this specific piece of music?*". Furthermore, we must also be aware that accuracy scores themselves may not be informative about "true" underlying performance [10,24]. Indeed, within the related field of recommender systems (which has some clear overlap with MIR), it has been observed that incrementing accuracy scores does not, in itself, lead to improvements in user experience [19].

Of course, a user-specific beat annotation can be obtained without any beat tracking algorithm, by manually annotating the desired beat locations, e.g. using software such as Sonic Visualiser [8]. However, manually annotating beat locations is a laborious procedure even for skilled annotators [15]. An alternative is to leverage multiple beat interpretations from a beat tracking algorithm, and then provide users with a range of solutions to choose from [12]. However, even with a large number of interpretations (which may be non-trivial and time-consuming to rank) there is no guarantee that the end-user's desired result will be among them, especially if the alternative interpretations are generated in a

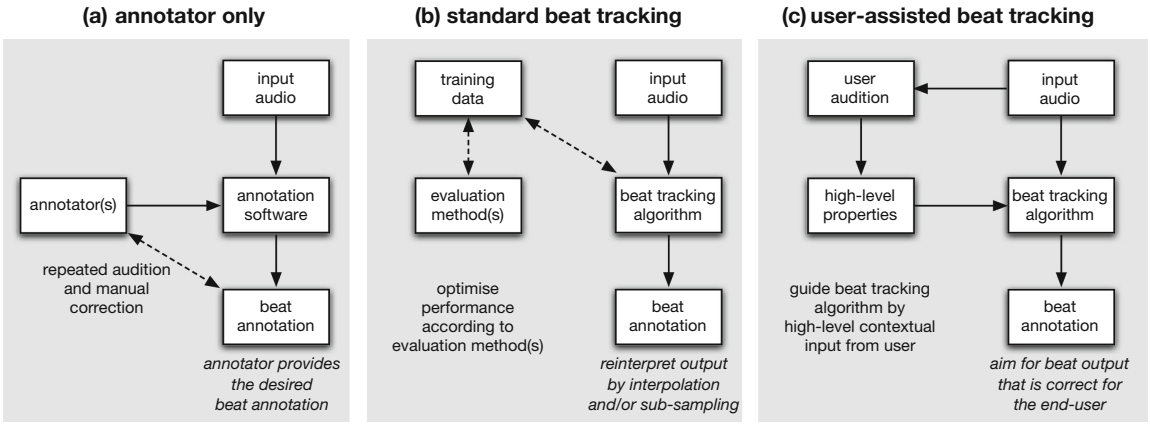


Fig. 1. Overview of different approaches to obtaining a desired beat annotation. (a) The user annotates the beat positions. (b) A beat tracking algorithm is used – whose performance has been optimised on annotated datasets. (c) Our proposed approach, where user input guides the beat tracking.

deterministic manner from a single beat tracking output, *e.g.*, by interpolation or sub-sampling.

In this paper, we propose an alternative formulation which could allow an end-user to drive how the beat tracking is undertaken. Our goal is to enable the user to rapidly arrive at the beat annotation suitable for their purposes with a minimal amount of interaction. Put another way, we envisage an approach to beat tracking where high-level contextual knowledge about a specific musical signal can be given by the user and reliably interpreted by the algorithm, without the need for extensive model training on annotated datasets, as shown in Fig. 1. In this sense, we put aside the concept of "universal" beat tracking models which target equal performance irrespective of the musical input signal, in favour of the more realistic goal of identifying different classes of the beat tracking problem, which require different beat tracking strategies. While the end goal of retrieving beat locations may be the same for fast-paced techno music and highly expressive classical guitar recordings, the assumptions about what constitutes the beat and how this can be extracted from audio signals are not. Conversely, constraints should not be placed on what musical content can be creatively re-purposed based on the limitations of MIR algorithms.

The long-term challenges of our approach are as follows: i) determining a low-dimensional parameterisation of the beat tracking space within which diverse, accurate solutions can be found in order to match different beat tracking conditions; ii) exposing these dimensions to end-users in a way that they can be easily understood; iii) providing an interpretable and understandable mapping between the user-input and the resulting beat annotation via the beat tracking algorithm; and finally iv) measuring the level of engagement among end-users who actively participate in the analysis of music signals.

Concerning the dimensions of beat tracking, it is well-understood that music of approximately constant (medium) tempo, with strong percussive content

(*e.g.*, pop, rock music) is straightforward to track. Beat tracking difficulty (both for computational approaches and human tappers) can be due to musical reasons and signal-based properties [13,15]. While it is somewhat nonsensical to consider a piece of music with "opposite" properties to the most straightforward case, it has been shown empirically that highly expressive music, without clear percussive content, is not well analysed even by the state of the art in beat tracking [5,15]. Successful tracking of such pieces should, in principle, require input features which can be effective in the absence of percussion and a tracking model which can rapidly adapt to expressive tempo variation. While recent work [4] sought to develop multiple beat tracking models, these were separately trained at the level of different databases rather than according to musical beat tracking conditions.

In our approach, we reexamine the functionality of the current state of the art in beat tracking, *i.e.*, the recurrent neural network approach of Böck et al. [5]. In particular, we devise a means to re-parameterise it so that it is adapted for highly expressive music. Based on an analysis of existing annotated datasets, we identify a set of musical stimuli we consider typical of highly challenging conditions, together with a parallel set of "easier" examples. We then conduct a small-scale listening experiment where participants are first asked to rate the perceptual difficulty of tapping the beat, and subsequently to rate the subjective quality of beat annotations given by the expressive parameterisation vs the default version. Our results indicate that listeners are able to distinguish easier from more challenging cases, and furthermore that they preferred the beat tracking output of the expressive-parameterised system to the default parameterisation for the highly expressive musical excerpts. In this sense, we seek to use the assessment of perceptual difficulty of tapping as a means to drive the manner in which the beats can be extracted from audio signals towards the concept of user-informed beat tracking. To complement our analysis, we explore the objective evaluation of the beat tracking model with both parameterisations.

The remainder of this paper is structured as follows. In Sect. 2 we detail the adaption of the beat tracking followed by the design of a small-scale listening experiment in Sect. 3. This is followed by results and discussion in Sect. 4, and conclusions in Sect. 5.

2 Beat Tracking System Adaptation

Within this work our goal is to include user input to drive how music signal analysis is conducted. We hypothesise that high-level contextual information which may be straightforward for human listeners to determine can provide a means to guide how the music signal analysis is conducted. For beat tracking, we established in Sect. 1 that for straightforward musical cases, the current state of the art [5] is highly effective. Therefore, in order to provide an improvement over the state of the art, we must consider the conditions in which it is less effective, in particular those displaying expressive timing. To this end, we first summarise the main functionality of the beat tracking approach of Böck et al., after which we detail how we adapt it.

The approach of Böck et al. [5] (originally presented in [6]) uses deep learning and is freely available within the madmom library [3]. The core of the beat tracking model is a recurrent neural network (RNN) which has been trained on a wide range of annotated beat tracking datasets to predict a beat activation function which exhibits peaks at likely beat locations. To obtain an output beat sequence, the beat activation function given by the RNN is post-processed by a dynamic Bayesian network (DBN) which is approximated by a hidden Markov model [20].

While it would be possible to retrain this model from scratch on challenging data, this has been partially addressed in the earlier multi-model approach of Böck et al. [4]. Instead, we reflect on the latter part of the beat tracking pipeline, namely how to obtain the beat annotation from the beat activation function. To this end, we address three DBN parameters: i) the minimum tempo in beats per minute (BPM); ii) the maximum tempo; and iii) the so-called "transition-λ" parameter which controls the flexibility of the DBN to deviate from a constant tempo[1]. Through iterative experimentation, including both objective evaluation on existing datasets and subjective assessment of the quality of the beat tracking output, we devised a new set of expressiveness-oriented parameters, which are shown, along with the default values in Table 1. More specifically, we first undertake a grid search across these three parameters on a subset of musical examples from existing annotated datasets for which the state of the art RNN is deemed to perform poorly, *i.e.*, by having an information gain lower than 1.5 bits [26]. An informal subjective assessment was then used to confirm that reliable beat annotations could be obtained from the expressive parameterisation.

Table 1. Overview of default and expressive-adapted parameters.

Parameter	Default	Expressive
Minimum Tempo (BPM)	55	35
Maximum Tempo (BPM)	215	135
Transition-λ (unitless)	100	10

As shown in Table 1, the main changes for the expressive model are a shift towards a slower range of allowed tempi (following evidence about the greater difficulty of tapping to slower pieces of music [7]), together with a lower value for the transition-λ. While the global effect of the transition-λ was studied by Krebs et al. [20], their goal was to find an optimal value across a wide range of musical examples. Here, our focus is on highly expressive music and therefore we do not need to pursue a more general solution. Indeed, the role of the expressive model is to function in precisely the cases where the default approach cannot.

[1] The probability of tempo changes varies exponentially with the negative of the "transition-λ", thus higher values of this parameter favour constant tempo from one beat to the next one [20].

3 Experimental Design

Within this paper, we posit that high-level user-input can lead to improved beat annotation over using existing state of the art beat tracking algorithms in a "blind" manner. In order to test this in a rigorous way, we would need to build an interactive beat tracking system including a user interface, and conduct a user study in which users could select their own input material for evaluation. However, doing so would require understanding which high-level properties to expose and how to meaningfully interpret them within the beat tracking system. To the best of our knowledge, no such experiment has yet been conducted, thus in order to gain some initial insight into this problem, we conducted a small-scale online listening experiment, which is split into two parts: **Part A** to assess the perceptual difficulty of tapping the beat, and **Part B** to assess the subjective quality of beat annotations made using the default parameterisation of the state of the art beat tracking system versus our proposed expressive parameterisation.

We use **Part A** as a means to simulate one potential aspect of high-level context which an end-user could provide: in this case, a choice over whether the piece of music is easy or difficult to tap along to (where difficulty is largely driven by the presence of expressive timing). Given this choice, **Part B** is used as the means for the end-user to rate the quality of the beat annotation when the beat tracking system has been parameterised according to their choice. In this sense, if a user rates the piece as "easy", we would provide the default output of the system, and if they rate it as "hard" we provide the annotation from the expressive parameterisation. However, for the purposes of our listening experiment, all experimental conditions are rated by all participants, thus the link between **Part A** and **Part B** is not explicit.

3.1 Part A

In the first part of our experiment, we used a set of 8 short music excerpts (each 15 s in duration) which were split equally among two categories: i) "easy" cases with near constant tempo in 4/4 time, with percussive content, and without highly syncopated rhythmic patterns; and ii) "hard" cases typified by the presence of high tempo variation and minimal use of percussion. The musical excerpts were drawn from existing public and private beat tracking datasets, and all were normalised to −3 dB.

We asked the participants to listen to the musical excerpts and to spontaneously tap along using the computer keyboard at what they considered the most salient beat. Due to the challenges of recording precise time stamps without dedicated signal acquisition hardware (*e.g.*, at the very least, a MIDI input device) the tap times of the participants were not recorded, however this was not disclosed. We then asked the participants to rate the difficulty they felt when trying to tap the beat, according to the following four options (Fig. 2):

- Low - *I could easily tap the beat, almost without concentrating*
- Medium - *It wasn't easy, but with some concentration, I could adequately tap the beat*

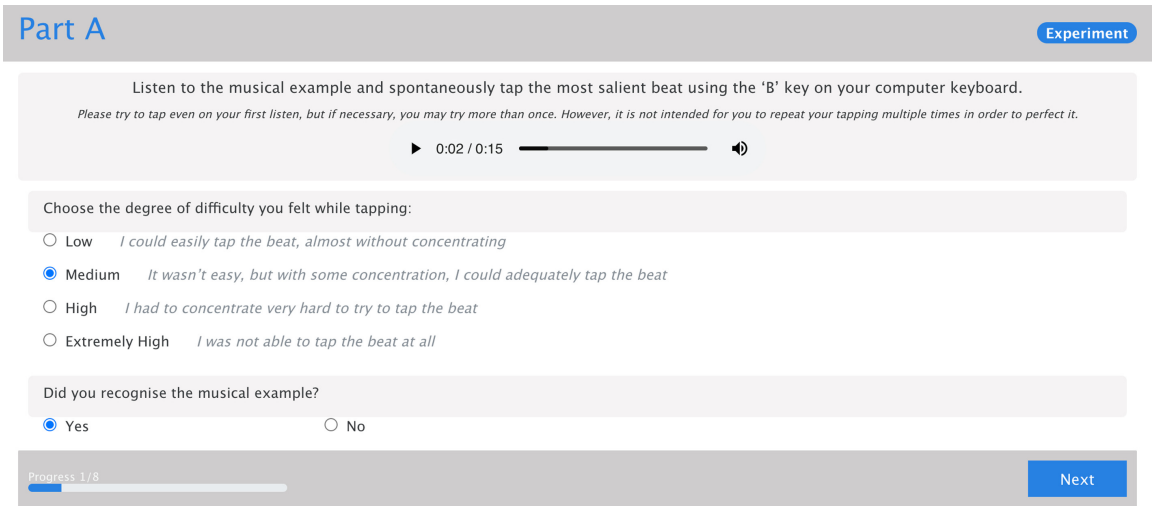


Fig. 2. Listening experiment - graphical interface of Part A.

– High - *I had to concentrate very hard to try to tap the beat*
– Extremely high - *I was not able to tap the beat at all.*

Our hypothesis for **Part A** is that participants would consistently rate those drawn from the "easy" set as having Low or Medium difficulty, whereas those from the "hard" should be rated with High or Extremely High difficulty.

3.2 Part B

Having completed **Part A**, participants then proceeded to **Part B** in which they were asked to judge the subjective quality of beat annotations (rendered as short 1 kHz pulses) mixed with the musical excerpts. The same set of musical excerpts from **Part A** were used, but they were annotated in three different ways: i) using the *default* parameterisation of the Böck et al. RNN approach from the madmom library [3]; ii) using our proposed *expressive* parameterisation (as in Table 1); and iii) a control condition using a completely *deterministic* beat annotation, *i.e.*, beat times at precise 500 ms intervals without any attempt to track the beat of the music. In total, this created a set of $8 \times 3 = 24$ musical excerpts to be rated, for which participants were asked to: *Rate the overall quality of how well the beat sequence corresponds to the beat of the music* (Fig. 3).

For this question, a 5-point Likert-type item was used with (1) on the left hand side corresponding to "Not at all" and (5) corresponding to "Entirely" on the right hand side. Our hypothesis for **Part B** was that for the "hard" excerpts, the annotations of the expressively-parameterised beat tracker would be preferred to those of the default approach, and for all musical excerpts that the deterministic condition would be rated the lowest in terms of subjective quality. In this part of the experiment we draw inspiration from evaluation of automatic musical accompaniment driven by real-time beat tracking where our three conditions of: default, expressive, and deterministic can be deemed similar

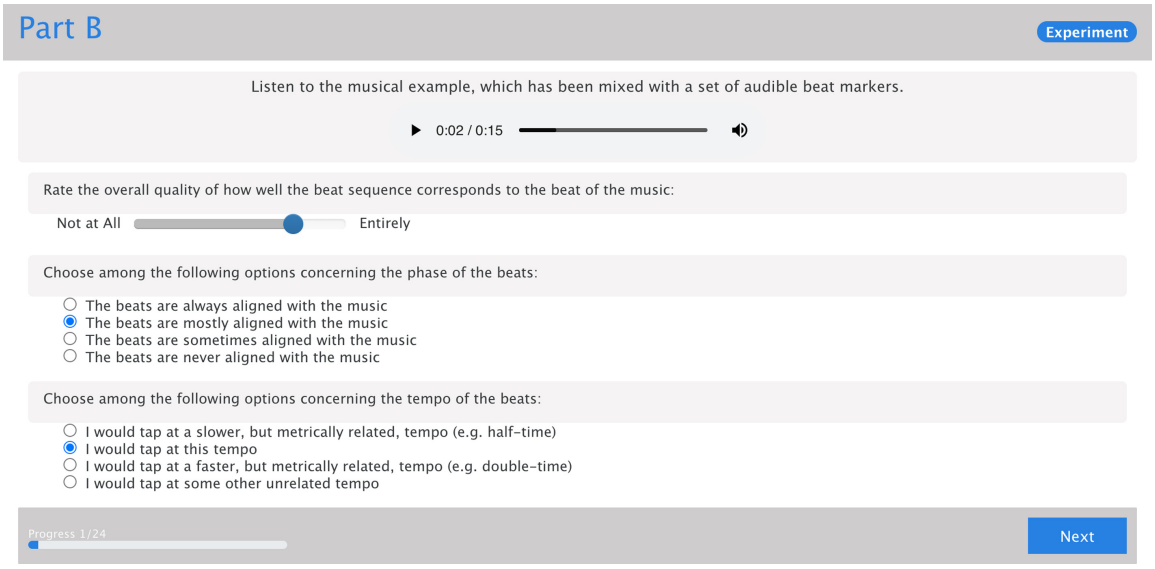


Fig. 3. Listening experiment - graphical interface of Part B.

to the use of a beat tracking system, a human tapper, and a quantised beat sequence, by Stowell et al. [23].

3.3 Implementation

The experiment was built using HTML5 and Node.js and run online within a web browser, where participants were recruited from the student body of the University of Porto and the wider research network of the Sound and Music Computing Group at INESC TEC. Within the experimental instructions, all participants were required to give their informed consent to participate, with the understanding that any data collected would be handled in an anonymous fashion and that they were free to withdraw at any time without penalty (and without their partial responses being recorded). Participants were asked to provide basic information for statistical purposes: sex, age, their level of expertise as a musician, and experience in music production.

All participants were encouraged to take the experiment in a quiet environment using high quality headphones or loudspeakers, and before starting, they were given the opportunity to set the playback volume to a comfortable level. Prior to the start of each main part of the experiment, the participants undertook a compulsory training phase in order to familiarise themselves with the questions. To prevent order effects, each participant was presented with the musical excerpts in a different random order. In total, the test took around 30 min to complete.

4 Results and Discussion

4.1 Listening Experiment

A total of 10 listeners (mean age: 31, age range: 23–43) participated in the listening test, 9 of whom self-reported amateur or professional musical proficiency.

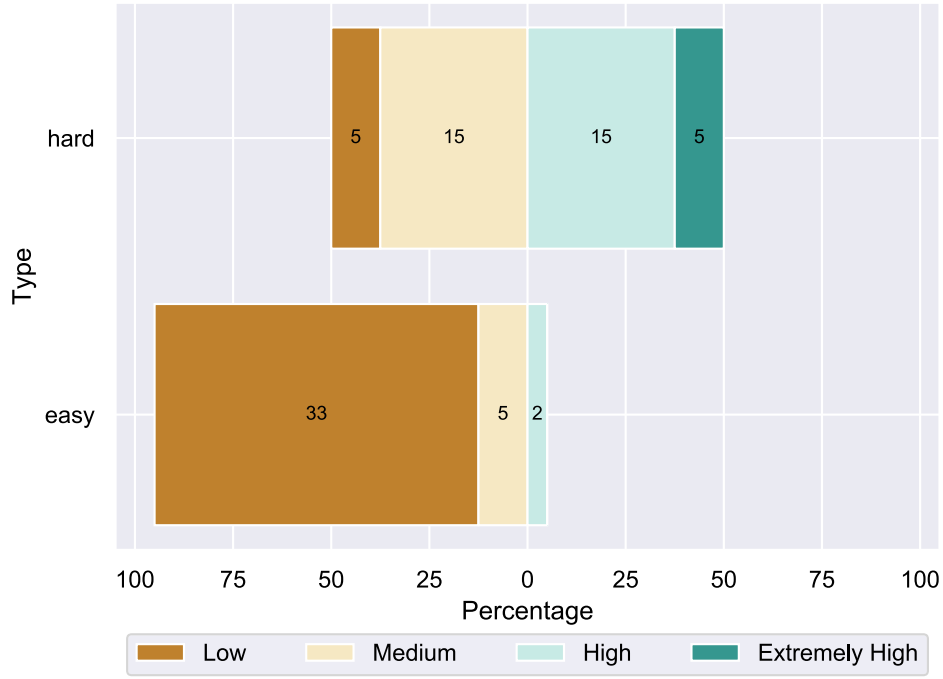


Fig. 4. Subjective ratings of the difficulty of beat tapping.

For **Part A**, we obtained 40 ratings for each stimuli group "easy" and "hard", according to the frequency distribution shown in Fig. 4. The most frequent rating for the first group was "low" (82.5%), followed by the "medium" rating (12.5%). For the "hard" group, a symmetrical rating was obtained: the adjacent ratings "medium" and "high" (37.5% each), complemented by the more extreme ratings "low" and "extremely high" (12.5% each). A Mann-Whitney test showed that there was a statistically significant difference between the ratings for both groups, with $p < 0.001$.

From these results we interpret that there was greater consistency in classifying the "easy" excerpts as having low difficulty, with only two excerpts rated above "medium", than for the "hard" excerpts which covered the entire rating scale from low to extremely difficult, albeit with the majority of ratings being for medium or high difficulty. We interpret this greater variability in the rating of difficulty of tapping to be the product of two properties of the participants: their expertise in musical performance and/or their familiarity with specific pieces. Moreover, we can observe a distinction between the understanding of the perceptual difficulty in tapping on the part of the participant and the presence of expressive timing in the musical excerpts; that experienced listeners

may not have difficulty in tapping along with a piece of expressive music for which they knew well. Thus, for expert listeners it may be more reasonable to ask a direct question related to the presence of expressive timing, while the question of difficulty may be more appropriate for non-expert listeners who might lack familiarity with the necessary musical terminology.

For **Part B**, we again make the distinction between the ratings of the "easy" and the "hard" excerpts. A Kruskal-Wallis H test showed that there was a statistically significant difference between the three models (*expressive*, *default* and *deterministic*): $\chi^2(2) = 87.96$, $p < 0.001$ for "easy" excerpts, $\chi^2(2) = 70.71$, $p < 0.001$ for "hard" excerpts. A post-hoc analysis performed with the Dunn test with Bonferroni correction showed that all the differences were statistically significant with $p < 0.001/3$ (except for the pair *default–expressive* under the "easy" stimuli, for which identical ratings were obtained). A descriptive summary of the ratings (boxplot with scores overlaid) for each type of stimuli, and under the three beat annotation conditions are shown in Fig. 5.

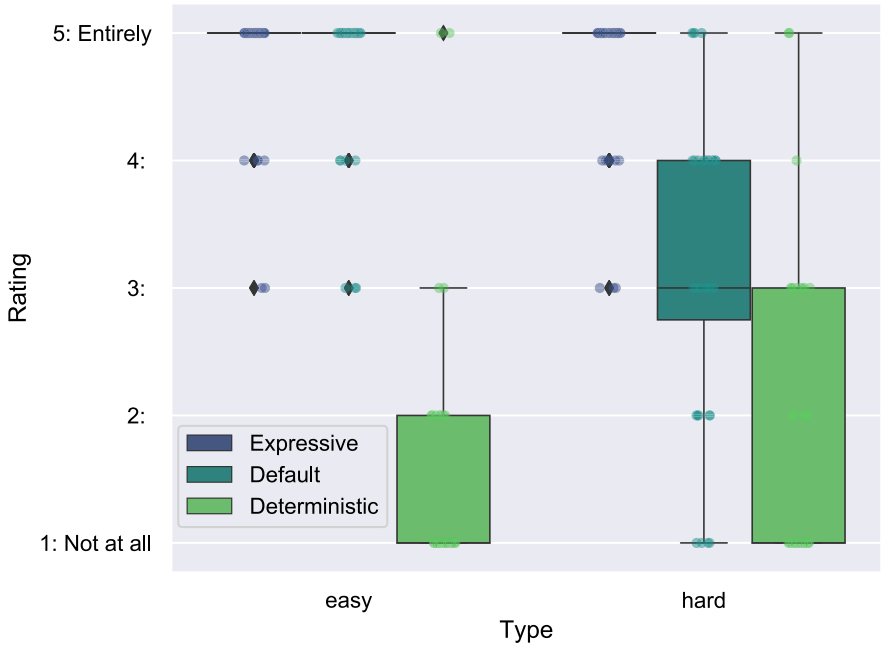


Fig. 5. Subjective ratings of the quality of the beat annotations.

The main results from Part B are as follows. For the "easy" excerpts there is no difference in performance for the *default* and *expressive* parameterisations of the beat tracking model, both of which are rated with high scores indicating high quality beat annotations from both systems. We contrast this with the ratings of the *deterministic* output (which should bear no meaningful relationship to the music) and which are rated toward the lower end of the scale. From these results we can infer that the participants were easily able to distinguish accurate beat annotations and deliberately inaccurate annotations, a result which is consistent with the well-known Beat Alignment Test [17]. Concerning the ability of the

expressively parameterised model to achieve such high ratings, we believe that this was due to very clear information concerning the beat in the beat activation functions from the RNN, and thus there was no alternative"expressive" path for this model to follow.

Conversely, the ratings of the "hard" excerpts show a different picture. Here, the ratings of the expressively-parameterised model are similar to the "easy" excerpts, but the ratings of the *default* model [3] are noticeably lower. This suggests that the participants, in spite of their reported higher perceptual difficulty in tapping the beat, were able to reliably identify the accurate beat predictions of the *expressive* model over those of the *default* model. It is noteworthy that the ratings of the *deterministic* approach are moderately higher for the "hard" excerpts compared to the "easy" excerpts. Given the small number of samples and participants for this experiment, we should not draw strong conclusions about this difference, but for highly expressive pieces, the *deterministic* beats may have inadvertently aligned with the music in brief periods compared to the "easy" excerpts, which may have been unrelated in a more obvious way to listeners.

4.2 Beat Tracking Accuracy

In addition to reporting on the listening experiment whose focus is on subjective ratings of beat tracking, we also examine the difference in objective performance of using the *default* and *expressive* parameterisations of the beat tracking model. Given the focus on challenging excerpts for beat tracking, we initially focus on the SMC dataset [15]. It contains 217 excerpts, each of 40 s in duration. Following the evaluation methods described in [10] we select the following subset: F-measure, CMLc, CMLt, AMLc, AMLt, and the Information Gain (D) to assess performance. In Tables 2, we show the recorded accuracy on the SMC for both the default and expressive parameterisations. Note, for the default model we use the version in the madmom library [3] which has been exposed to this material during training (via cross-validation), hence the accuracy scores are slightly higher than those in [5] where 8-fold cross validation was used. In addition to showing the performance of each parameterisation we also show the theoretical upper limit achievable by making a perfect choice (by a hypothetical end-user) among the two parameterisations. Since multiple evaluation scores are reported, and there is no accepted single metric to use within the beat tracking community, we make the optimal choice per excerpt according to each individual evaluation metric.

From Table 2, we see that for all the evaluation methods, with the exception of the Information Gain (D), the default parameterisation outperforms the expressive one. This result is not unexpected result as the SMC dataset is not entirely comprised of highly expressive musical material. We consider the more important result to be the potential for our *expressive* parameterisation to track those excerpts for which the *default* approach fails. To this end, the increase of approximately 10% points across each of the evaluation methods demonstrates how these two different parameterisations can provide greater coverage of the dataset.

Table 2. Overview of beat tracking performance on the SMC dataset [15] comparing the default and expressive parameters together with upper limit on performance.

	F-measure	CMLc	CMLt	AMLc	AMLt	D
Default [3]	0.563	0.350	0.472	0.459	0.629	1.586
Expressive	0.540	0.306	0.410	0.427	0.565	1.653
Optimal Choice	0.624	0.456	0.611	0.545	0.703	1.830

While the SMC dataset is well-known for containing a high proportion of challenging material, we also believe that it is worthwhile to explore the effectiveness of our method on other musical material. Since the expressive parameterisation should only be effective when applied to music with a slow average tempo and high expression, the gains on datasets comprised primarily of pop or rock music will be much lower. In addition, many of the existing beat tracking datasets have been used to train the approach of Böck et al. [5] and thus cannot provide insight into the effectiveness of our approach on truly unseen data. To this end, we make use of a more recently annotated dataset which was used in the 2017 IEEE Signal Processing Cup (SP Cup) [18]. While the dataset is quite small, containing 98 excerpts of 30 s it was compiled in a community-driven fashion where teams participating in the competition selected the audio material and annotated it themselves. In line with the competitive element of the SP Cup many teams chose to submit challenging musical excerpts. On this basis, we believe it represents a highly appropriate choice for additional validation of our approach. A summary of the results containing the same three conditions: default, expressive, and the optimal choice between the two, is shown in Table 3.

Table 3. Overview of beat tracking performance SP Cup dataset [18] comparing the default and expressive parameters together with upper limit on performance.

	F-measure	CMLc	CMLt	AMLc	AMLt	D
Default [3]	0.833	0.660	0.687	0.846	0.877	2.968
Expressive	0.783	0.564	0.581	0.805	0.826	2.955
Optimal Choice	0.860	0.733	0.762	0.873	0.897	3.062

Contrasting the results in Tables 2 and 3 we can observe a similar pattern of lower overall performance for the expressive approach compared the default parameterisation. However, once again, the optimal choice between the two provides a notable improvement (of up to 7% points) depending on the evaluation method. Given the improvement under both presented datasets we believe this supports the need for different parameterisations to tackle different types of musical content, a concept related to Collins' discussion of "style-specific" beat tracking [9]. In addition, it suggests that training a classifier to choose between expressive and non-expressive pieces would be a promising area for future work.

4.3 Individual Example

While results shown in Tables 2 and 3 focus more on the global effect of these different paramaterisations across entire datasets, it is important to consider the practical impact at the level of individual musical excerpts. In this section we consider an annotation workflow perspective, which might rely on the correction of an automatic annotation of the beat of a piece of music, as opposed to completely annotating a piece by hand. In this context, we contend that an informed choice of how to first estimate the beat automatically may have a significant impact in terms of the subsequent work required to obtain an output which is acceptable for the end-user, i.e. by inserting, deleting, and shifting the automatically estimated beats.

To this end, we focus on one specific example within the Hainsworth dataset [14]; an excerpt from the composition "Evocaciòn" by Jose Luis Merlin. It is a solo piece for classical guitar which features extensive *rubato* and as such can be considered one of the more challenging pieces within the dataset. In the absence of any other musical instruments, together with the clear guitar plucking technique, this piece is rather a paradox since it is quite straightforward for onset detection, but notoriously difficult for beat tracking. The challenge lies not in the ability to precisely identify where in time the notes are played, but to decode which of these onsets correspond to the beat over a highly variable underlying tempo. To explore this specific musical excerpt in greater detail, we contrast the outputs of the default and expressive parameterisations together with the ground truth annotation (taken from the supplementary material from [2]) in Fig. 6.

As can be seen from the figure, the output of the expressive parameterisation (in the bottom plot) is much closer to the ground truth annotations than the default (in the top plot). Across this 30 s section there are just 6 beats in need of correction for the expressive output, with no fewer than 18 for the default output. The number of atomic operations to correct each annotation can be broken down as follows: 13 shifts and 6 deletions for the default output vs. 3 shifts and 3 deletions for the expressive output. Taking into account the number of annotations in this excerpt, the amount of editing effort required to converge on the ground-truth annotation is even more illustrative: 21% of the expressive beats output vs 61% of the default beat outputs. Thus, from the user (annotator) perspective, it is clearly more efficient to correct the expressive output.

In this example, we have explicitly used the ground truth as a means to illustrate the fewer number of errors made by the expressive parameterisation. However, when such ground truth annotations exist, the need for automatic analysis is negated. Yet, in real-world uses, where there is no ground truth, we would replace this visual comparison with an interactive process whereby the user verifies the output of the algorithm by listening and iterative adjustment. The number of edit operations to achieve the desired output indicates the amount of interactions between the user and the beat-tracking system, and can thus provide a direct indicator of the impact of user-informed beat tracking in the annotation workflow.

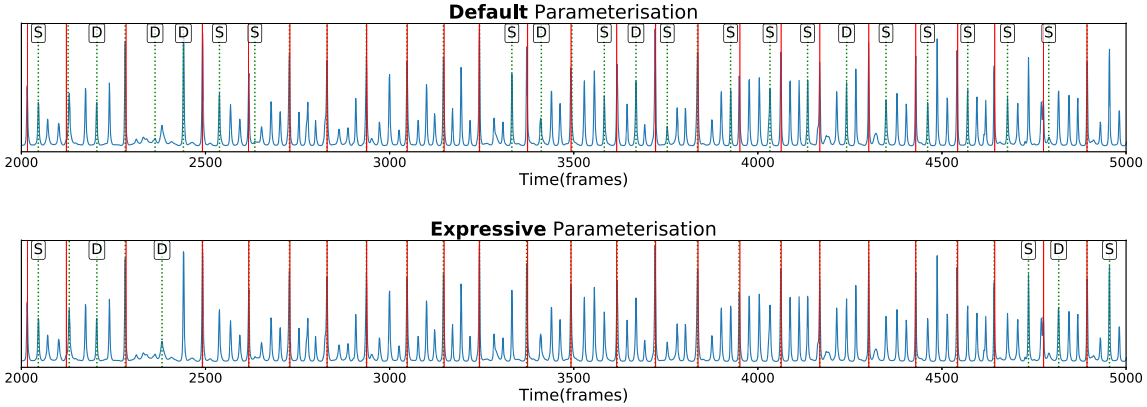


Fig. 6. Comparison of different beat tracking outputs. The blue solid line indicates the beat activation function given by the Böck et al. approach [5]. The vertical red solid lines show the ground truth annotations. The vertical green dashed lines show: the default output (top) and the expressive output (bottom). The incorrect beat outputs are labelled with the required operations (**D**elete, **S**hift, **I**nsert) to correct the annotation. The temporal axis represents frames at a rate of 100 frames per second. (Color figure online)

5 Conclusions

In this paper we have sought to open the discussion about the potential for user-input to drive how MIR analysis is performed. Within the context of beat tracking, we have demonstrated that it is possible to reparameterise an existing state-of-the-art approach to provide better beat annotations for highly expressive music, and furthermore, that the ability to choose between the default and expressive parameterisation can provide significant improvements on very challenging beat tracking material. We emphasise that the benefit of the expressive model was achieved without the need for any retraining of the RNN architecture, but that the improvement was obtained by reparameterisation of the DBN tracking model which performs inference on the prediction of the RNN.

To obtain some insight into how user input could be used for beat tracking, we simulated a scenario where user decisions about perceptual difficulty of tapping could be translated into the use of a parameterisation for expressive musical excerpts. We speculate that listener expertise as well as familiarity may play a role in lowering the perceived difficulty of otherwise challenging expressive pieces. Our intention is to further investigate the parameters which can be exposed to end-users, and whether different properties may exist for expert compared to non-expert users. Despite the statistical significance of our results, we acknowledge the small-scale nature of the listening experiment, and we intend to expand both the number of musical excerpts uses as well as targeting a larger group of participants to gain deeper insight into the types of user groups which may emerge. Towards our long-term goal, we will undertake an user study not only to understand the role of beat tracking for creative MIR, but also to assess the level of engagement when end-users are active participants who guide the analysis.

Acknowledgments. This work is supported by Portuguese National Funds through the FCT-Foundation for Science and Technology, I.P., under the grant SFRH/BD/120383/2016 and the project IF/01566/2015.

References

1. Andersen, K., Knees, P.: Conversations with expert users in music retrieval and research challenges for creative MIR. In: Proceedings of the 17th International Society for Music Information Retrieval Conference, pp. 122–128 (2016)
2. Böck, S., Davies, M.E.P., Knees, P.: Multi-task learning of tempo and beat: learning one to improve the other. In: Proceedings of the 20th International Society for Music Information Retrieval Conference, pp. 486–493 (2019)
3. Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., Widmer, G.: Madmom: a new python audio and music signal processing library. In: Proceedings of the 2016 ACM Multimedia Conference, pp. 1174–1178 (2016). https://doi.org/10.1145/2964284.2973795
4. Böck, S., Krebs, F., Widmer, G.: A multi-model approach to beat tracking considering heterogeneous music styles. In: Proceedings of the 15th International Society for Music Information Retrieval Conference, pp. 603–608 (2014)
5. Böck, S., Krebs, F., Widmer, G.: Joint beat and downbeat tracking with recurrent neural networks. In: Proceedings of the 17th International Society for Music Information Retrieval Conference, pp. 255–261 (2016)
6. Böck, S., Schedl, M.: Enhanced beat tracking with context-aware neural networks. In: Proceedings of the 14th International Conference on Digital Audio Effects, pp. 135–139 (2011)
7. Bååth, R., Madison, G.: The subjective difficulty of tapping to a slow beat. In: Proceedings of the 12th International Conference on Music Perception and Cognition, pp. 82–55 (2012)
8. Cannam, C., Landone, C., Sandler, M.: Sonic visualiser: an open source application for viewing, analysing, and annotating music audio files. In: Proceedings of the 18th ACM International Conference on Multimedia, pp. 1467–1468 (2010). https://doi.org/10.1145/1873951.1874248
9. Collins, N.: Towards a style-specific basis for computational beat tracking. In: Proceedings of the 9th International Conference on Music Perception and Cognition (ICMPC), pp. 461–467 (2006)
10. Davies, M.E.P., Böck, S.: Evaluating the evaluation measures for beat tracking. In: Proceedings of the 15th International Society for Music Information Retrieval Conference, pp. 637–642 (2014)
11. Desain, P., Honing, H.: Does expressive timing in music performance scale proportionally with tempo? Psychol. Res. **56**(4), 285–292 (1994). https://doi.org/10.1007/BF00419658
12. Goto, M., Yoshii, K., Fujihara, H., Mauch, M., Nakano, T.: Songle: a web service for active music listening improved by user contributions. In: Proceedings of the 12th International Society for Music Information Retrieval Conference, pp. 311–316 (2011)
13. Grosche, P., Müller, M., Sapp, C.: What makes beat tracking difficult? A case study on chopin mazurkas. In: Proceedings of the 11th International Society for Music Information Retrieval Conference, pp. 649–654 (2010)
14. Hainsworth, S.: Techniques for the Automated Analysis of Musical Audio. Ph.D. thesis, University of Cambridge (2004)

15. Holzapfel, A., Davies, M.E.P., Zapata, J.R., Oliveira, J., Gouyon, F.: Selective sampling for beat tracking evaluation. IEEE Trans. Audio Speech Lang. Process. **20**(9), 2539–2548 (2012). https://doi.org/10.1109/TASL.2012.2205244
16. Humphrey, E.J., Turnbull, D., Collins, T.: A brief review of creative MIR. In: Late-breaking Demo Session of the 14th International Society for Music Information Retrieval Conference (2013)
17. Iversen, J.R., Patel, A.D.: The Beat Alignment Test (BAT): surveying beat processing abilities in the general population. In: Proceedings of the 10th International Conference on Music Perception and Cognition, pp. 465–468 (2010)
18. Jin, C.T., Davies, M.E.P., Campisi, P.: Embedded systems feel the beat in new orleans: highlights from the IEEE signal processing cup 2017 student competition [SP Competitions]. IEEE Signal Process. Mag. **34**(4), 143–170 (2017). https://doi.org/10.1109/MSP.2017.2698075
19. Konstan, J.A., Riedl, J.: Recommender systems: from algorithms to user experience. User Model. User-Adap. Inter. **22**(1), 101–123 (2012). https://doi.org/10.1007/s11257-011-9112-x
20. Krebs, F., Böck, S., Widmer, G.: An efficient state space model for joint tempo and meter tracking. In: Proceedings of the 16th International Society for Music Information Retrieval Conference, pp. 72–78 (2015)
21. Moelants, D., McKinney, M.: Tempo perception and musical content: what makes a piece fast, slow or temporally ambiguous? In: Proceedings of the 8th International Conference on Music Perception and Cognition, pp. 558–562 (2004)
22. Serra, X., et al.: Roadmap for music information research (2013), Creative Commons BY-NC-ND 3.0 license, ISBN: 978-2-9540351-1-6
23. Stowell, D., Robertson, A., Bryan-Kinns, N., Plumbley, M.D.: Evaluation of live human-computer music-making: quantitative and qualitative approaches. Int. J. Hum. Comput. Stud. **67**(11), 960–975 (2009). https://doi.org/10.1016/j.ijhcs.2009.05.007
24. Sturm, B.L.: Classification accuracy is not enough. J. Intell. Inf. Syst. **41**(3), 371–406 (2013). https://doi.org/10.1007/s10844-013-0250-y
25. Urbano, J., Schedl, M., Serra, X.: Evaluation in music information retrieval. J. Intell. Inf. Syst. **41**(3), 345–369 (2013). https://doi.org/10.1007/s10844-013-0249-4
26. Zapata, J.R., Holzapfel, A., Davies, M.E.P., Oliveira, J.L., Gouyon, F.: Assigning a confidence threshold on automatic beat annotation in large datasets. In: Proceedings of the 13th International Society for Music Information Retrieval Conference, pp. 157–162 (2012)

electronics

MDPI

Article

User-Driven Fine-Tuning for Beat Tracking

António S. Pinto [1,*], Sebastian Böck [2], Jaime S. Cardoso [1] and Matthew E. P. Davies [3]

[1] INESC TEC, Centre for Telecommunications and Multimedia, 4200-465 Porto, Portugal; jaime.cardoso@inesctec.pt
[2] enliteAI, 1000-1901 Vienna, Austria; s.boeck@enlite.ai
[3] Centre for Informatics and Systems, Department of Informatics Engineering, University of Coimbra, 3030-290 Coimbra, Portugal; mepdavies@dei.uc.pt
* Correspondence: antonio.s.pinto@inesctec.pt

Abstract: The extraction of the beat from musical audio signals represents a foundational task in the field of music information retrieval. While great advances in performance have been achieved due the use of deep neural networks, significant shortcomings still remain. In particular, performance is generally much lower on musical content that differs from that which is contained in existing annotated datasets used for neural network training, as well as in the presence of challenging musical conditions such as *rubato*. In this paper, we positioned our approach to beat tracking from a real-world perspective where an end-user targets very high accuracy on specific music pieces and for which the current state of the art is not effective. To this end, we explored the use of targeted fine-tuning of a state-of-the-art deep neural network based on a very limited temporal region of annotated beat locations. We demonstrated the success of our approach via improved performance across existing annotated datasets and a new annotation-correction approach for evaluation. Furthermore, we highlighted the ability of content-specific fine-tuning to learn both what is and what is not the beat in challenging musical conditions.

Keywords: beat tracking; transfer learning; user adaptation

Citation: Pinto, A.S.; Böck, S.; Cardoso, J.S.; Davies, M.E.P. User-Driven Fine-Tuning for Beat Tracking. *Electronics* **2021**, *10*, 1518. https://doi.org/10.3390/electronics10131518

Academic Editors: Alexander Lerch and Peter Knees

Received: 25 May 2021
Accepted: 18 June 2021
Published: 23 June 2021

1. Introduction

A long-standing area of investigation in music information retrieval (MIR) is the computational rhythm analysis of musical audio signals. Within this broad research area, which incorporates many diverse facets of musical rhythm including onset detection [1], tempo estimation [2] and rhythm quantisation [3], sits the foundational task of musical audio beat tracking. The goal of beat tracking systems is commonly stated as inferring and then tracking a quasi-regular pulse so as to replicate the way a human listener might subconsciously tap their foot in time to a musical stimulus [4–6]. However, the pursuit of computational beat tracking is not limited to emulating an aspect of human music perception. Rather, it has found widespread use as an intermediate processing step within larger scale MIR problems by allowing the analysis of harmony [7] and long-term structure [8] in "musical time" thanks to beat-synchronous processing. In addition, the imposition of a beat grid on a musical signal can enable the extraction and understanding of expressive performance attributes such as microtiming [9]. Furthermore, within creative applications of MIR technology, the accurate extraction of the beat is of critical importance for synchronisation and thus plays a pivotal role in automatic DJ mixing between different pieces of music [10], as well as the layering of music signals for mashup creation [11]. In particular for musicological and creative applications, the need for very high accuracy is paramount as the quality of the subsequent analysis and/or creative musical result will depend strongly on the accuracy of the beat estimation.

From a technical perspective, computational approaches to musical audio beat tracking (as with many MIR tasks) have undergone a profound transformation due to the prevalence

of deep neural networks. While numerous traditional approaches to beat tracking exist, it can be argued that they follow a largely similar set of processing steps: (i) the calculation of a time–frequency representation such as a short-time Fourier transform (STFT) from the audio signal; (ii) the extraction of one or more mid-level representations from the STFT, e.g., the use of complex spectral difference [12] or other so-called "onset detection functions" [13], whose local maxima are indicative of the temporal locations of note onsets; and (iii) the simultaneous or sequential estimation of the periodicity and phase of the beats from this onset detection function (or an extracted discrete sequence of onsets) with techniques such as autocorrelation [14], comb filtering [15], multi-agent systems [16,17] and dynamic programming [18]. The efficacy of these traditional approaches was demonstrated via their evaluation on annotated datasets, many of which were small and not publicly available.

By contrast, more recent supervised deep learning approaches sharply diverge from this formulation in the sense that they start with, and explicitly depend on, access to large amounts of annotated training data. The prototypical deep learning approach, perhaps best typified by Böck and Schedl [19], formulates beat tracking as a sequential learning problem of binary classification through time, where beat targets are rendered as impulse trains. The goal of a beat-tracking deep neural network, typically by means of recurrent and/or convolutional architectures, is to learn to predict a beat activation function from an input representation (either the audio signal itself or a time–frequency transformation), which closely resembles the target impulse train. While in some cases, it can be sufficient to employ thresholding and/or peak-picking to obtain a final output sequence of beats from this beat activation function, the de facto standard is to use a dynamic Bayesian network (DBN) [20] approximated by a hidden Markov model (HMM) [21] for inference, which is better able to contend with spurious peaks or the absence of reliable information. Given this explicit reliance on annotated training data, together with the well-known property of neural networks to "overfit" to training data, great care must be taken when evaluating these systems to ensure that all test data remain unseen by the network in order to permit any meaningful insight into the generalisation capabilities.

Following this data-driven formulation, the state of the art in beat tracking has improved substantially over the last 10 years, with the most recent approaches using temporal convolutional networks [22], achieving accuracy scores in excess of 90% on diverse annotated datasets comprised of rock, pop, dance and jazz musical excerpts [23–26]. Yet, in spite of these advances, several challenges and open questions remain. Deep learning methods are known to be highly data-sensitive [27]. The knowledge they acquire is directly linked both to the quality of the annotated data and the scope of musical material to which they have been exposed. In this sense, it is hard to predict the efficacy of a beat tracking system when applied to "unfamiliar" (i.e., outside of the dataset) musical material; indeed, even state-of-the-art systems that perform very well on Western music have been shown to perform poorly on non-Western music [9]. Likewise, given the arduous nature of the manual annotation of beat locations for the creation of annotated datasets, there is an implicit bias towards more straightforward musical material, e.g., with a roughly constant tempo, 4/4 metre, and the presence of drums [28,29]. In this way, more challenging musical material, e.g., containing highly expressive tempo variation, non-percussive content, changing metres, etc., is under-represented, and its relative scarcity in annotated datasets may contribute to poorer performance. Furthermore, the great majority of annotated datasets comprise musical excerpts of up to one minute in duration, meaning that the ability of these systems to track entire musical pieces in a structurally consistent manner is largely unknown.

The scope and motivation for this paper were to move away from the notion of targeting and then reporting high (mean) accuracy across existing annotated datasets and instead to move towards the real-world use of beat tracking systems by end-users on specific musical pieces. More specifically, we investigated what to do when even the state of the art is not effective and very high accuracy is required, i.e., when the extraction of the beat is used to drive higher level musicological analysis or creative musical repurposing.

Faced with this situation, currently available paths of action include: (i) the end-user performing manual corrections to the beat output or even resorting to a complete re-annotation by hand, which may be extremely time-consuming and labour-intensive; (ii) the use of some high-level parameterisation of the algorithm in terms of an expected tempo range and initial phase [16,30]; or (iii) adapting some more abstract parameters that could permit greater flexibility in tracking tempo variation [31]. While at first sight promising, this high-level information may only help in a very limited way: if the musical content is very expressive, then knowing some initial tempo might not be useful later on in the piece. Likewise, if the model is unable to make reliable predictions of the beat-like structure given the presence of different signal properties (e.g., timbre), then this user provided information may only be useful in very localised regions.

In light of these limitations, we proposed a user centric approach to beat tracking in which a very limited amount of manual annotation by a hypothetical end-user is used to fine-tune an existing state-of-the-art system [22] in order to adapt it to the specific properties of the musical piece being analysed. In essence, we sought to leverage the general musical knowledge of a beat-tracking system exposed to a large amount of training data and then to recalibrate the weights of the network so that it can rapidly learn how to track the remainder of the given piece of music to a high degree of accuracy. A high-level overview of this concept is illustrated in Figure 1. However, in order for this to be a practical use case, it is important that the fine-tuning process be computationally efficient and not require specialist hardware, i.e., that the fine-tuning can be completed in a matter of seconds on a regular personal computer. To demonstrate the validity of our approach, we showed the improvement over the current state of the art offered by our fine-tuning approach on existing datasets and by the specific examples, demonstrating that our approach can learn what is the beat, and also what is not the beat. In addition, we investigated the trade-off between learning the specific properties of a given piece and forgetting more general information. In summary, the main contributions of this work were: (i) to reformulate the beat-tracking problem to target high accuracy in individual challenging pieces where the current state-of-the-art is not effective; (ii) to introduce the use of in situ fine-tuning over a small annotated region as a straightforward means to adapt a state-of-the-art beat-tracking system so that it is more effective for this type of content; and (iii) to conduct a detailed beat-tracking evaluation from an annotation-correction perspective, which demonstrates and quantifies the set of steps required to transform an initial estimate of the beat into a highly accurate output.

The remainder of this paper is structured as follows: In Section 2, we discuss our approach to fine-tuning in the context of existing work on transfer learning in MIR. In Section 3, we provide a high-level overview of the state-of-the-art beat-tracking system used as the basis for our approach and then detail the fine-tuning in Section 4. In Sections 5 and 6, we present a detailed evaluation employing a widely used method along with a recent evaluation approach specifically designed to address the extent of user correction. Finally, in Section 7, we discuss the implications and limitations of our work and propose promising areas of future research.

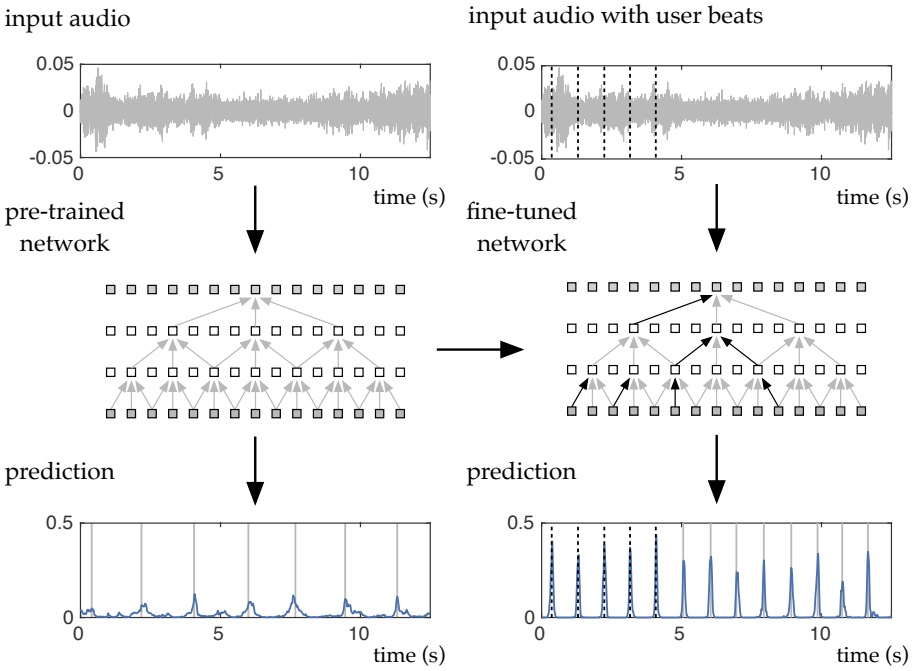


Figure 1. Overview of our proposed approach. The left column shows an audio input passed through a deep neural network (for consistency with our approach, this is a temporal convolutional network), which produces a weak beat activation function and erroneous beat output. The right column shows the same audio input, but here, a few beat annotations are provided as the means to fine-tune the network—with the black arrows implying the modification of some of the weights of the network. This results in a much clearer beat activation function and an accurate beat-tracking output.

2. Low-Data Learning Strategies

Data scarcity represents a major bottleneck for machine learning in general, but particularly for deep learning. Within the musical audio domain, data curation is often hindered by the laborious and expensive human annotation process, subjectivity and content availability limitations due to copyright issues, thus making the field of MIR an interesting use-case for machine learning strategies to address low-data regimes [32]. Following success in the research domains of computer vision and natural language processing, a wide range of approaches have been proposed to overcome this limitation in the audio domain. In this paper, we focused on one such approach, *transfer learning*, through which knowledge gained during training in one type of problem is used to train another related task or domain [33]. Leveraging previously acquired knowledge and avoiding a cold-start (i.e., training "from scratch"), it can enable the development of accurate models in a cost-effective way.

Early approaches to transfer learning in MIR were based on the use of pretrained models on large datasets for feature extraction and have been proposed for tasks such as genre classification and auto-tagging [34], speech/music classification or music emotion prediction [35]. A different methodology is the use of pretrained weights as an initialization for the parameters of the downstream model. This technique, known as *fine-tuning*, proposes the subsequent retraining of certain parts of the network by defining which weights to "unfreeze" while retaining the existing knowledge in the "frozen" components. This parameter transfer learning approach has been used for the adaptive generation of rhythm microtiming [36] and for beat tracking, as a way to transfer the knowledge of a network trained on popular music into tracking beats in Greek folk music [37].

Another strategy for low-data regimes is known as *few-shot learning*, which aims at generalizing from only a few examples [38]. Both paradigms have been studied for music classification tasks [39]. Lately, the association between both approaches has become widespread, with transfer learning techniques being widely deployed in few-shot classi-

fication, achieving high performance with a simplicity that has made fine-tuning the de facto baseline method for few-shot learning [40], in what is known as *transductive transfer learning* [33].

Within the context of musical audio beat tracking, we employed fine-tuning not for the adaption to a new task per se, but rather to new content within the same task. In formal terms, this can be considered *sequential inductive transfer learning*. Our approach differs from that of Fiocchi et al. [37] since we targeted a kind of controlled overfitting to a specific piece of music rather than a collection of musical excerpts in a given style. In this sense, our approach bears some high-level similarity to the use of "bespoke" networks for audio source separation [41]. While the need to rely on some minimal annotation effort could be seen as an inefficiency in a processing pipeline, which, in many MIR contexts, is fully automatic [42], our approach may offer the means to address subjectivity in beat perception via personalised analysis.

3. Baseline Beat-Tracking Approach

A key motivating factor and contribution of this work is to look beyond what is possible with the current state of the art in beat tracking, and hence to explore fine-tuning as a means for content-specific adaptation. To this end, we restricted the scope of this work to an explicit extension of the most recent state-of-the-art approach [22], and thus used this as a baseline on which to measure improvement.

The baseline approach uses multi-task learning for the simultaneous estimation of beat, downbeat and tempo. The core of the approach is a temporal convolutional network (TCN), which was first used for beat tracking only in [43], and then expanded to predict both tempo and beat [44]. Compared to previous recurrent architectures for beat tracking (e.g., [45]), TCNs have the advantage that they retain the high parallelisation property of convolutional neural networks (CNNs), and therefore can be trained more efficiently over large training data [43]. With the long-term goal of integrating in situ fine-tuning within a user based workflow for a given piece of music, we considered this aspect of efficiency to be particularly important, and this therefore formed a secondary motivation to extend the TCN-based approach.

To provide a high-level overview of this approach ahead of the discussion of fine-tuning, and to enable this paper to be largely self-contained, we now summarise the main aspects of the processing pipeline, network architecture and training procedure. For complete details, see [22].

Pre-processing: Given a mono audio input signal, sampled at 44.1 kHz, the input representation is a log magnitude spectrogram obtained with a *Hann* window of 46.4 ms (2048 samples) and a hop length of 10 ms. Subsequently, a logarithmic grouping of frequency bins with 12 bands per octave gives a total of 81 frequency bands from 30 Hz up to 17 kHz.

Neural network: The neural network was comprised of two stages: a set of three convolutional and max pooling layers followed by a TCN block. The goal of the convolutional and max pooling layers was to learn a compact intermediate representation from the musical audio signal, which could then be passed to the TCN as the main sequence learning model. The shapes of the three convolutional and max pooling layers were as follows: (i) 3×3 followed by 1×3 max pooling; (ii) 1×10 followed by 1×3 max pooling; and (iii) 3×3 again with 1×3 max pooling. A dropout rate of 0.15 was used with the exponential linear unit (ELU) as the activation function.

This compact intermediate representation was then fed into a TCN block that operated noncausally (i.e., with dilations spanning both forwards and backwards in time). The TCN block was composed of two sets of geometrically spaced dilated convolutions over eleven layers with one-dimensional filters of size five. The first of the dilations spanned the range of 2^0 up to 2^{10} frames and the second at twice this rate. The feature maps of the two dilated convolutions were concatenated before spatial dropout (with a rate of 0.15) and the ELU as activation function. Finally, in order to keep the output dimensionality

of the TCN layer consistent, these feature maps were combined with a 1×1 convolution. Within the multitask approach (and unlike the simultaneous estimation in [45]), the beat and downbeat targets were separate, each produced by a sigmoid on a fully connected layer. The tempo classification output was produced by a softmax layer. In total, twenty filters were learned within this network, giving approximately 116 k weights. A graphical overview of the network is given in Figure 2.

Training: The network was trained on the following six reference datasets, which totalled more than 26 h of musical material: *Ballroom* [26,46], *Beatles* [24], *Hainsworth* [23,44], *HJDB* [45,47], *Simac* [48] and *SMC* [28]. In order to account for gaps in the distribution of the tempi of these datasets, a data augmentation strategy was adopted, by which the training data were enlarged by a factor of 10, by varying the overlap rate of the frames of the STFT (and hence the tempo) and by sampling from a normal distribution with the 5% standard deviation around the annotated tempo and updating the beat, downbeat and tempo targets accordingly. Furthermore, to account for the high imbalance between positive and negative examples (i.e., that frames labelled as beats occurred much less often than nonbeat frames), the beat and downbeat targets were widened by ± 2 frames and weighted by 0.5 and 0.25 as they diverged from the central beat frame.

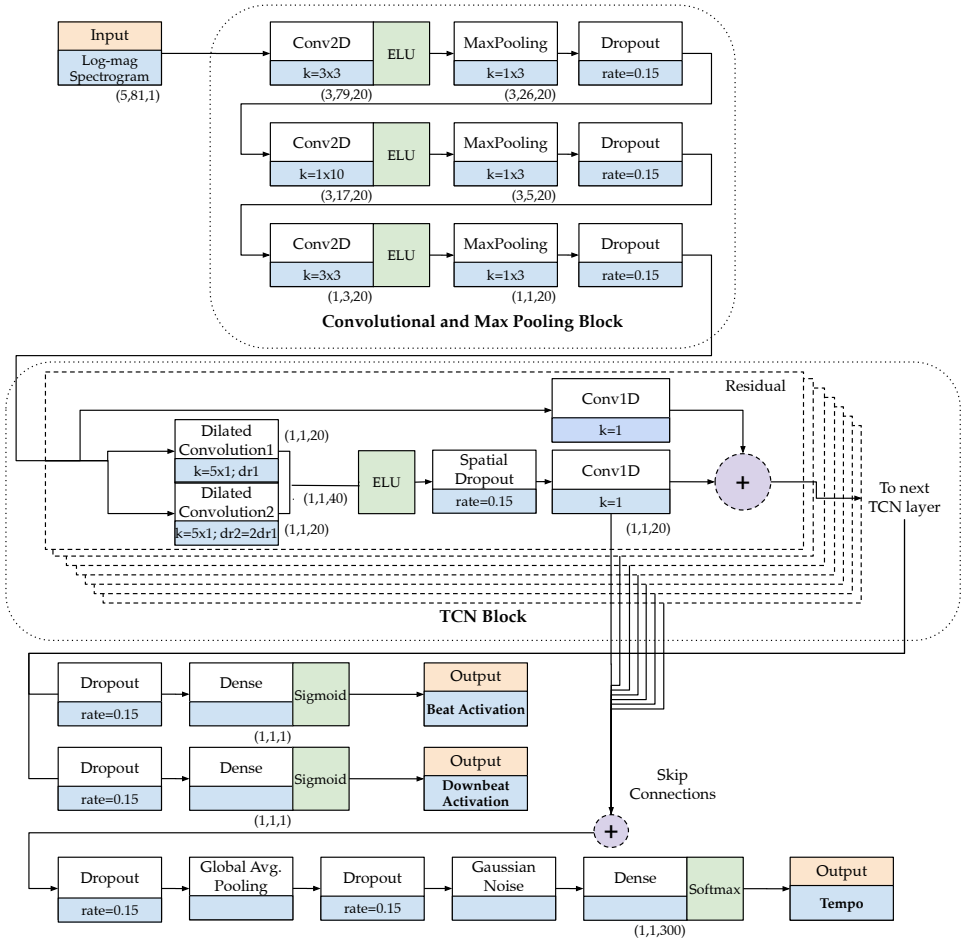


Figure 2. Overview diagram of the architecture of the baseline beat-tracking approach.

The training was conducted using eight-fold cross validation (6 folds for training, 1 fold for validation, and 1 fold held-back for testing), with excerpts from each dataset uniformly distributed across the folds. A maximum of 200 training epochs per fold were used with a learning rate of 0.002, which was halved after no improvement in the validation loss for 20 epochs, and early stopping was activated with no improvement after 30 epochs.

The *RAdam* optimiser followed by *lookahead optimization* were used with a batch size of one and gradient clipping at a norm of 0.5.

Postprocessing: To obtain the final output, the beat activation and downbeat activations were combined and passed as the input to a dynamic Bayesian network approximated via an HMM [45], which simultaneously decoded the beat times and labels corresponding to metrical position (i.e., where the all beats labelled 1 were downbeats). However, given only the beat activation function, it was possible to use the beat-only HMM for inference [21].

4. Fine-Tuning

Departing from the network architecture described above, we now turn our attention toward how we could adapt it to successfully analyse very challenging musical pieces. It is important to restate that our interest was specifically in musical content for which the current state-of-the-art approach is not effective and for which high accuracy is desired by some end-user. Within this scenario, it is straightforward to envisage that some form of user input could be beneficial to guide the estimation of the beat.

In a broad sense, our strategy was to take advantage of the transferability of features in neural networks [49], in effect to leverage the global knowledge about beat tracking from the baseline approach and the datasets upon which it has been trained, and to recalibrate it to fit the musical properties of a given new piece. By connecting this concept of transferability with an end-user who actively participated in the analysis and a prototypical beat annotation workflow, we formulated the network adaption as a process of fine-tuning based on a small temporal region of manually annotated beat positions. From the user perspective, this implies a small annotation effort to mark a few beats by hand, and then using this information as the basis for updating the weights of the baseline network such that the complete piece can be accurately analysed with minimal further user interaction.

Within this paper, our primary interest was to understand the viability of this approach, rather than testing it in real-world conditions. To this end, we simulated the annotation effort of the end-user by using ground truth annotations over a small temporal region and examining how well the adapted network could track the remainder of the piece. From a technical perspective, we began with a pretrained model from the baseline approach described in the previous section. Then, for a given musical excerpt (unseen to the pretrained model), we isolated a small temporal region (nominally near the start of the excerpt), which we set to be 10 s in duration, and retrieved the corresponding ground truth beat annotations. Together, these three components formed the basis of our fine-tuning approach, as illustrated in Figure 1. In devising this approach, we focused on: (i) how to parameterise the fine-tuning; (ii) when to stop the fine-tuning; and (iii) how to cope with the very limited amount of new information provided by the small temporal region.

Fine-tuning parameterisation: The first consideration in our fine-tuning approach was to examine which layers of the baseline network to update. It is commonplace in transfer learning to freeze all but the last layers of the network [50]. However, in our context, one important means for adapting the network resides in modelling how the beat is conveyed within the log magnitude spectrogram itself (i.e., unfamiliar musical timbres such as the human voice). To this end, we allowed all the layers of the network to be updated by the fine-tuning process. Since our focus in this paper was restricted to beat tracking, we masked the losses for the tempo and downbeat tasks. From a practical perspective, this also means that we did not require downbeat or tempo annotations across the 10 s temporal region. Concerning the parameterisation of the fine-tuning, we followed common practice in transfer learning and reduced the learning rate, setting it to 0.0004 (i.e., one fifth of the rate used in the baseline).

Stopping criteria: The next area was to address when to stop fine-tuning. In more standard approaches for training deep neural networks, e.g., our baseline approach, cross-fold validation is used with the validation loss driving the adjustment of the learning rate and the execution of early stopping. In our approach, if we were to use the entire 10 s region for training, then it would be difficult to exercise control over the extent of

the network adaption. Using a small, fixed number of epochs might leave the network essentially unchanged after fine-tuning, and by contrast, allowing a large number of epochs might cause the network to overfit in an adverse manner. Furthermore, the hypothetically optimal number of epochs is likely to vary based on the musical content being analysed. Faced with this situation, we elected to split the 10 s region into two adjacent, disjoint, 5 s regions, using one for training and the other for validation. In this way, we created a validation loss that we could monitor, but at the expense of reducing the amount of information available for updating the weights. We set the maximum number of epochs to fifty and reduced the learning rate by a factor of two when there was no improvement in the validation loss for at least five epochs, and we stopped training when the validation loss plateaued for five epochs.

Learning from very small data: The final area for consideration in our approach relates to strategies to contend with the very limited amount of information in the 5 s temporal region used for training, which may amount to as few as 10 annotated beat targets. Given our interest in challenging musical content (which is typically more difficult to annotate [28]), we should consider the fact that these observable annotations may be poorly localised, and furthermore that the tempo may vary throughout the piece in question. To help contend with poor localisation, we used a broader target widening strategy than the baseline approach, expanding to three adjacent frames on either side of each beat location, with decreasing weights of 0.5, 0.25 and 0.125, from the closest to the farthest frame. On the issue of tempo variability, we reused the same data augmentation from the baseline approach: altering the frame overlap rate by sampling from a normal distribution with a 5% standard deviation from the local tempo (calculated by means of the median inter-beat interval across the annotated region).

In summary, when considering each of these steps, we believe that our fine-tuning formulation was quite general and could be applied to any pretrained network for beat tracking, and was thus not specific to the TCN-based approach we chose to extend.

5. User Workflow-Based Evaluation

In recent work [51], we introduced a new approach for beat tracking evaluation, which formulates it from a user workflow perspective. Within this paper, it formed a key component within our evaluation, and thus, to make this paper self-contained, we provide a full description here.

We posed the problem in terms of the effort required to transform a sequence of beat detections such that they maximise the well-known F-measure calculation when compared to a sequence of ground truth annotations. By viewing the evaluation from a transformation perspective, we implicitly used the commonly accepted definition for the similarity between two objects (i.e., the beat annotations and the beat detections) in the field of information retrieval [52], in effect to answer: *How difficult is it to transform one into the other?* By combining this perspective with an informative visualisation, we sought to support a better qualitative understanding of beat-tracking algorithms, and thus, we adopted the same approach in this work. Within our current work, we did not attempt to explicitly incorporate this evaluation method within our fine-tuning approach via backpropagation, rather we used it only as a guide to interpret the end result.

In musical audio analysis, the manual alteration of automatically detected time-precise musical events such as onsets [53] or beats [54] is an onerous process. In the case of musical beat tracking, the beat detections may be challenging due to the underlying difficulty of the musical material, but the correction process can be achieved using two simple editing operations: insertions and deletions—combined with repeated listening to audible clicks mixed with the input. The number of insertions and deletions correspond to counts of *false negatives* and *false positives*, respectively, and form part of the calculation of the F-measure. While this is routinely used in beat tracking (and many other MIR tasks) to measure accuracy, we can also view it in terms of the effort required to transform an initial set of

beat detections to a final desired result (e.g., a ground truth annotation sequence). In this way, a high F-measure would imply low effort in manual correction and vice versa.

In practice, correcting beat detections often relies on a third operation: the *shifting* of poorly localised individual beats. This shifting operation is particularly relevant when correcting tapped beats, which can be subject to human motor noise (i.e., random disturbances of signals in the nervous system that affect motor behaviour [55]), as well as jitter and latency during acquisition. Under the logic of the F-measure calculation, shifting beat detections that fall outside tolerance windows are effectively counted twice: as a false positive *and* a false negative. We argue that for beat tracking evaluation, this creates a modest, but important, disconnect between common practice in annotation correction and a widely used evaluation method. On this basis, we recommend that the single operation of shifting should be prioritised over a deletion followed by an insertion.

In parallel, we also devised a straightforward calculation for the *annotation efficiency* based on counting the number of shifts, insertions and deletions. In our approach, we weighted these different operations equally. Although valid in an abstract way, in practice, the real cost of such operations depends on the annotation workflow of the user, in which we included the supporting editing software tool (e.g., in a particular software, evenly spaced events could be annotated by providing only an initial beat position, the tempo in BPM and the duration, while in another software, each beat event may have to be annotated individually).

We provide an open-source Python implementation (Available at https://github.com/MR-T77/ShiftIfYouCan (accessed on 25 May 2021)), which graphically displays the minimum set and type of operations required to transform a sequence of initial beat detections in such a way as to maximise the F-measure when comparing the transformed detections against the ground truth annotations. It is important to note that our goal was *not* to transform the beat detections such that they were absolutely identical to the ground truth (although such transformations are theoretically possible), but rather to perform as few operations as possible to ensure F = 1.00, subject to a user defined tolerance window.Nevertheless, in its current implementation, the assignment of estimated events (beats) to one of the possible operations was performed by a locally greedy matching strategy. In future work, we will explore the use of global optimization using graphs, as in [56].

We now specify the main steps in the calculation of the transformation operations:

1. Around each ground truth annotation, we created an *inner tolerance window* (set to ± 70 ms) and counted the number of *true positives* (unique detections), t^+;
2. We marked each matching detection and annotation pair as "accounted for" and removed them from further analysis. All remaining detections then became candidates for *shifting* or *deletion*;
3. For each remaining annotation:
   (a) We looked for the closest "unaccounted for" detection within an *outer tolerance window* (set to ± 1 s), which we used to reflect a localised working area for manual correction;
   (b) If any such detection existed, we marked it as a shift along with the required temporal correction offset;
4. After the analysis of all "unaccounted for" annotations was complete, we counted the number of shifts, s;
5. Any remaining annotations corresponded to false negatives, f^-, with leftover detections marked for deletion and counted as false positives, f^+.

To give a measure of annotation efficiency, we adapted the evaluation method in [16] to include the shifts:

$$ae = t^+ / (t^+ + s + f^+ + f^-). \quad (1)$$

Reducing the inner tolerance window transforms true positives into shifts and thus sends t^+ and hence *ae* to zero. In the limit, the modified detections are then identical to the target sequence.

To allow for metrical ambiguity in beat tracking evaluation, it is common to create a set of variations of the ground truth by interpolation and subsampling operations. In our implementation, we flipped this behaviour, and instead created variations of the detections. In this way, we could couple a global operation applied to all detections (e.g., interpolating all detections by a factor of two), with the subsequent set of local correction operations; whichever variation has the highest annotation efficiency represents the shortest path to obtaining an output consistent with the annotations.

The fundamental difference of our approach compared to the standard F-measure is that we viewed the evaluation from a user workflow perspective, and essentially, *we shifted if we could*. By recording each individual operation, we could count them for evaluation purposes, as well as visualising them, as shown in Figure 3, which contrasts the use of the original beat detections compared to the double variation of the beats. The example shown is from the composition *Evocaciòn* by Jose Luis Merlin. It is a solo piece for classical guitar, which features extensive *rubato* and is among the more challenging pieces in the *Hainsworth* dataset [23]. By inspection, we can see the original detections were much closer to the ground truth than the offbeat or double variation. They required just 2 shifts and 1 insertion, compared with 12 shifts, 3 insertions and 1 deletion for the offbeat variation (without any valid detection), and 3 shifts and 12 deletions for the double variation, corresponding to very different annotation efficiency scores on the analysed excerpt: 0.8, 0.0 and 0.4, respectively.

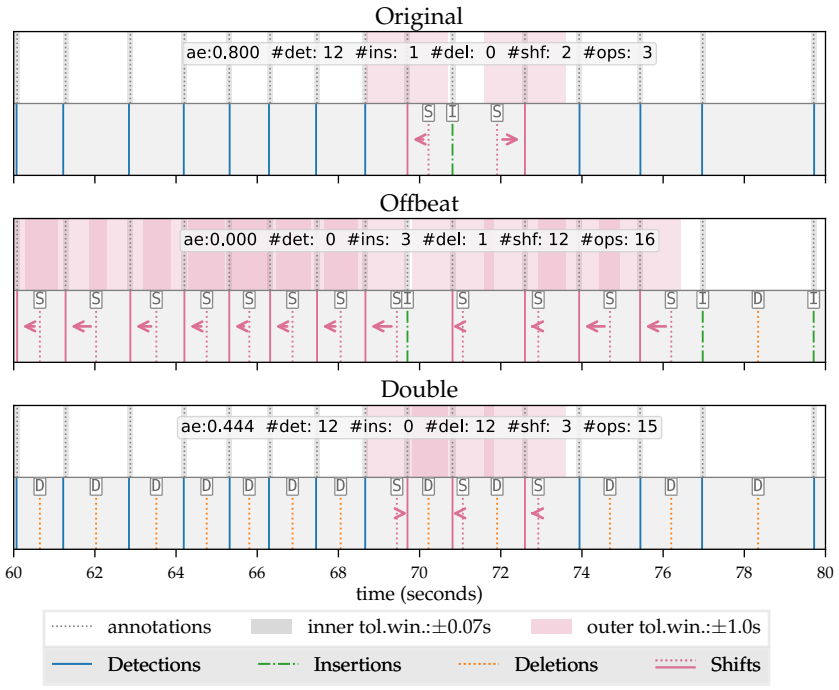


Figure 3. Visualisation of the operations required to transform beat detections to maximise the F-measure when compared to the ground truth annotations for the period from 60–80 s, of *Evocaciòn*. (Top) *Original* beat detections vs. ground truth annotations. (Middle) *Offbeat*—180 degrees out of phase from the original beat locations—variation of beat detections vs. ground truth annotations. (Bottom) *Double*—beats at two times the original tempo—beat detections vs. ground truth annotations. The inner tolerance window is overlaid on all annotations, whereas the outer tolerance window is only shown for those detections to be shifted.

The precise recording of the set of individual operations allowed an additional deeper evaluation, which could indicate precisely which operations were most beneficial and in

which order. For the F-measure, shifts were always more beneficial than the isolated insertions or deletions, but for other evaluation methods, i.e., those that measure continuity, the temporal location of the operation may be more critical. By viewing the evaluation from a transformation perspective combined with an informative visualisation, we hope our implementation can contribute to a better qualitative understanding of beat-tracking algorithms.

6. Experiments and Results

In this section, we start by detailing the design of our experimental setup, after which we measured the performance on a set of existing annotated datasets. We then explored the impact of fine-tuning in two specific highly challenging musical pieces. Finally, we investigated the presence and extent of catastrophic forgetting. When combined, we considered that these multiple aspects constituted a rigorous analysis of our proposed approach.

6.1. Experimental Setup

As detailed in Section 4, our fine-tuning process relied on a short annotated region for training and an additional region of equal duration for validation. We reiterate that in this work where we sought to broadly investigate the validity of fine-tuning over a large amount of musical material, we simulated the role of the end-user, and to this end, we obtained these annotated regions from existing beat tracking datasets rather than direct user input. While the duration and location of these regions within the musical excerpt were somewhat arbitrary compared to a practical use case with an end-user, for this evaluation, we chose them to be 5 s in duration each and adjacent to one another starting from the first annotated beat position per excerpt. By choosing the first beat annotation as opposed to the beginning of the excerpt, we could avoid any degenerate training that might otherwise arise if no musical content occurred within the first 10 s of an excerpt (e.g., a long nonmusical intro). For the purposes of evaluation, the impact of this configuration of fine-tuning across the early part of the excerpt had the advantage that it was straightforward to trim these regions to which the network had been exposed prior to inference with the HMM and then offset the annotations accordingly. In this way, we could contrast the performance of the fine-tuned version with the baseline model [22] without any impact of the sharp peaks in the beat activation functions across the training region. Note that due to the removal of the training and validation regions when evaluating, the results we obtained were not directly comparable to those in [22], which used the full-length excerpts. To summarise, our goal in formulating the evaluation was to see the extent to which the adaptation of the network over a short region near the start of each excerpt was reflected through the rest of the piece.

6.2. Performance Across Common Datasets

While our long-term interest in this work was towards a workflow setting with an end-user, we believe that it is valuable to first investigate the effectiveness of our approach on existing datasets and hence to obtain insight into its validity over a wide range of musical material. To this end, we used four datasets: two from the cross-fold validation training methodology in the baseline model [22]: the *SMC* dataset [28] and the *Hainsworth* dataset [23]; and two totally unseen by the original model: the *GTZAN* dataset [25,57], which was held back for testing, and the *TapCorrect* dataset [54], upon which the baseline model has never been evaluated. In terms of the musical make-up of these datasets, *Hainsworth* includes rock/pop, dance, folk, jazz, classical and choral. *SMC* contains classical, romantic, soundtracks, blues, chanson and solo guitar. *GTZAN* spans 10 genres, including: rock, disco, jazz, reggae, blues and classical. *TapCorrect* is comprised of mostly pop and rock music. Of particular note for the *TapCorrect* dataset is the fact that it contains entire musical pieces rather than the more customary use of excerpts from 30–60 s, and therefore, this could provide insight concerning the propagation of the acquired knowledge from the short training region over much longer durations. A summary of the datasets used is shown in Table 1. When performing fine-tuning on *SMC* and *Hainsworth*, we respected the original splits in the cross-fold validation in [22] and used the appropriate saved model

file, which was held out for testing. As stated above, the *GTZAN* dataset was not included in the splits for cross-validation, meaning we could not make a deterministic selection of which pretrained model to fine-tune. In the evaluation in [22], the final output per excerpt was obtained by predicting a beat activation function with the model from each fold of the cross-validation and then taking their temporal average (so-called "bagging") prior to inference with the HMM. While we could pursue this strategy here, it would involve fine-tuning eight separate times (once per fold) and therefore would significantly increase the computation time. Instead, we made a random selection among the trained models and only performed fine-tuning once. Informal evaluation over repeated runs revealed the specific choice of model to have little impact on the results.

Table 1. Overview of the datasets used for the evaluation.

Dataset	# Files	Full Length	Mean File Length
Hainsworth	222	3 h 19 m	53 s
SMC	217	2 h 25 m	40 s
GTZAN	999	8 h 18 m	30 s
TapCorrect	101	7 h 15 m	4 m 18 s

To measure performance across these datasets, we used the F-measure with the standard tolerance window of ±70 ms. The results for each dataset are shown in Table 2.

Table 2. Mean F-measure scores across datasets for the baseline and fine-tuning approaches.

Dataset	Baseline	Fine-Tuned
	F-Measure	F-Measure
Hainsworth	0.899	0.945
SMC	0.551	0.589
GTZAN	0.879	0.917
TapCorrect	0.911	0.941

Inspection of Table 2 demonstrates that the inclusion of fine-tuning exceeded the performance of the baseline state-of-the-art approach for all datasets—even accounting for the deterministic choice of region for fine-tuning. However, while some broad interpretation could be made by observing accuracy scores at the level of datasets, we could better understand the impact of the fine-tuning via a scatter plot of the baseline vs. the fine-tuned F-measure per excerpt and per dataset, as shown in Figure 4.

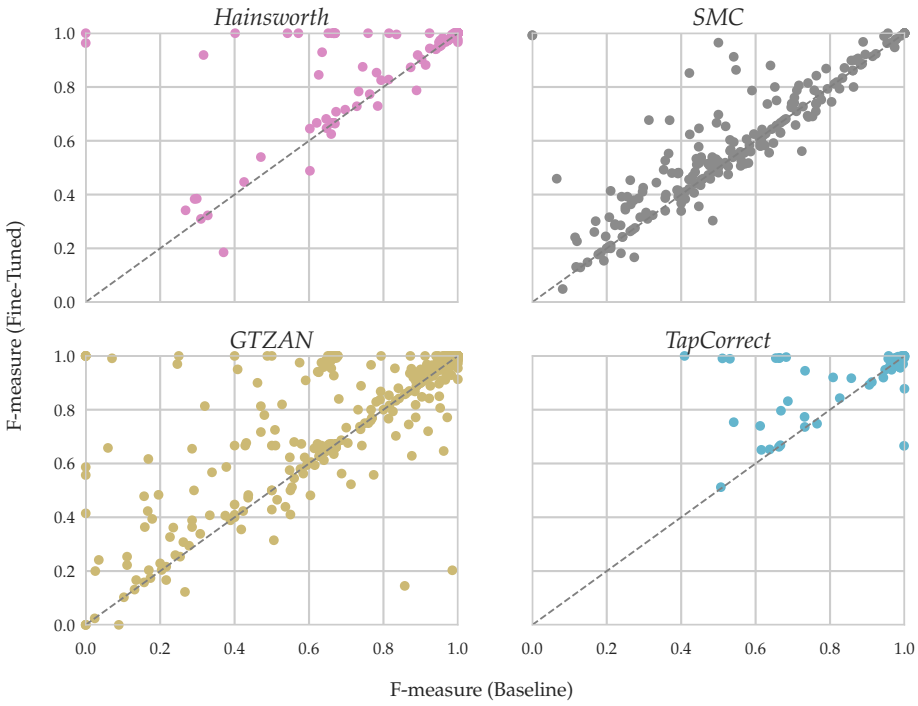


Figure 4. Comparison of the F-measure for the baseline and fine-tuning approaches on in-training datasets *Hainsworth* and *SMC* and out-of-training datasets *GTZAN* and *TapCorrect*.

To observe a positive impact of fine-tuning in the scatter plots, we looked for F-measure scores that are above the main diagonal, i.e., the F-measure per excerpt with fine-tuning improved over the baseline. Contrasting the scatter plots in terms of this behaviour, we observe that for *Hainsworth* and *TapCorrect*, very few pieces fall below the main diagonal, indicating that the fine-tuning was almost never worse. At this stage, it is worthwhile to reaffirm that if the performance was already very high for the baseline approach, then there was very limited scope for improvement with fine-tuning. Indeed, such cases fell outside our main use-case of interest, which was to consider what action to take when the state-of-the-art approach failed. In terms of the nature of the improvements, we can observe some explainable patterns. For example, those pieces for which the F = 0 for the baseline and F = 1 for the fine-tuning were almost certainly phase corrections from *offbeat* (i.e., out-of-phase) to *onbeat* (i.e., in-phase) at the annotated metrical level. Likewise, any improvement of F = 0.67 to F = 1 was very likely a correction in the choice of metrical level by doubling or halving, i.e., a change to the metrical level corresponding to twice or half the tempo, respectively. Alternatively, we can see that for those pieces that straddle the main diagonal, the impact of the fine-tuning is negligible. Finally, at the other end of the spectrum, we can observe that for *SMC* and *GTZAN*, there are at least some cases for which the fine-tuning negatively impacted performance. However, we should note that there are very few extreme outliers where it was catastrophically worse to fine-tune. Ultimately, the cases of most interest to us were those which sit on or close to the line F = 1 after fine-tuning, as these represent those for which there was the clearest benefit.

To obtain a more nuanced perspective, we reported the counts of all the operations necessary to calculate the annotation efficiency, namely the insertions, deletions and shifts required to transform a set of detections so as to maximize the F-measure. This information is displayed in Table 3. By contrasting the baseline and fine-tuned approaches, we see that across all datasets, fewer total editing operations were required. Indeed, per class of operation, the use of fine-tuning also resulted in fewer insertions, deletions and shifts. In this sense, we interpreted that the impact of fine-tuning was more pronounced than merely correcting the metrical level or phase of the detected beats. Thus, even accounting for the fact that, from a user perspective, each of these operations might not be equally

easy to perform, and a reduction across all operation classes highlighted the potential for the improved efficiency of an annotation-correction workflow.

Table 3. Global number of atomic edit operations: correct detections (#det), insertions (#ins), deletions (#del), shifts (#shf) and total edit operations (#ops) for the different test datasets.

Dataset	Model	#det	#ins	#del	#shf	#ops
Hainsworth	Baseline	16,498	923	455	837	2215
	Fine-Tuned	17,241	500	246	517	1263
SMC	Baseline	4593	810	1337	2457	4604
	Fine-Tuned	5028	670	1107	2162	3939
GTZAN	Baseline	33,505	3348	1132	2235	6715
	Fine-Tuned	35,403	1911	492	1774	4177
TapCorrect	Baseline	35,072	3285	1622	910	5817
	Fine-Tuned	36,659	2115	1236	493	3844

6.3. Impact on Individual Excerpts

In this section, we take a more direct look at the impact of fine-tuning by focussing on two specific pieces, a choral version of the song *Blue Moon*, taken from the *Hainsworth* dataset, and a full-length performance of the Heitor Villa-Lobos composition *Choros №1*, as performed by the Korean guitarist Kyuhee Park.

6.3.1. *Blue Moon*

Blue Moon (Excerpt Number 134 from the *Hainsworth* dataset [23]) is an *a cappella* performance and thus contains no drums or other musical instrumentation besides the voices of the performers. Nevertheless, the performance has a clear metrical structure driven not only by the lyrics and melody, but also the orchestration of different musical parts by the singers. On this basis, it represents an interesting case for further exploration, as choral music is well known to be extremely challenging for musical audio beat-tracking systems [28]. In Figure 5, we plot the log magnitude spectrogram with beat annotations overlaid as white dotted lines. As can be seen, there is very little high-frequency information with most energy concentrated under 4 kHz—and thus consistent with singing. In the middle plot, we can observe the beat activation function produced by the baseline approach together with the ground truth annotations. By inspection, we can see that the peaks of the beat activation function are very low, which is indicative of the low confidence of the baseline model in its output. Following the same strategy used for the evaluation across the datasets, we used the ground truth annotations and performed fine-tuning across the period in the first 10 s of the recording, validating on the first period of 5 s and training on the second period of 5 s, with the resulting beat activation function shown in the lowest plot of the figure. Contrasting the two beat activation functions, we can observe a profound difference. Once we allowed the network to adapt itself to the spectrotimbral properties of the beat structure of this specific piece, we can see a series of regular sharp peaks in the beat activation, which visually correspond to the overlaid manual annotations.

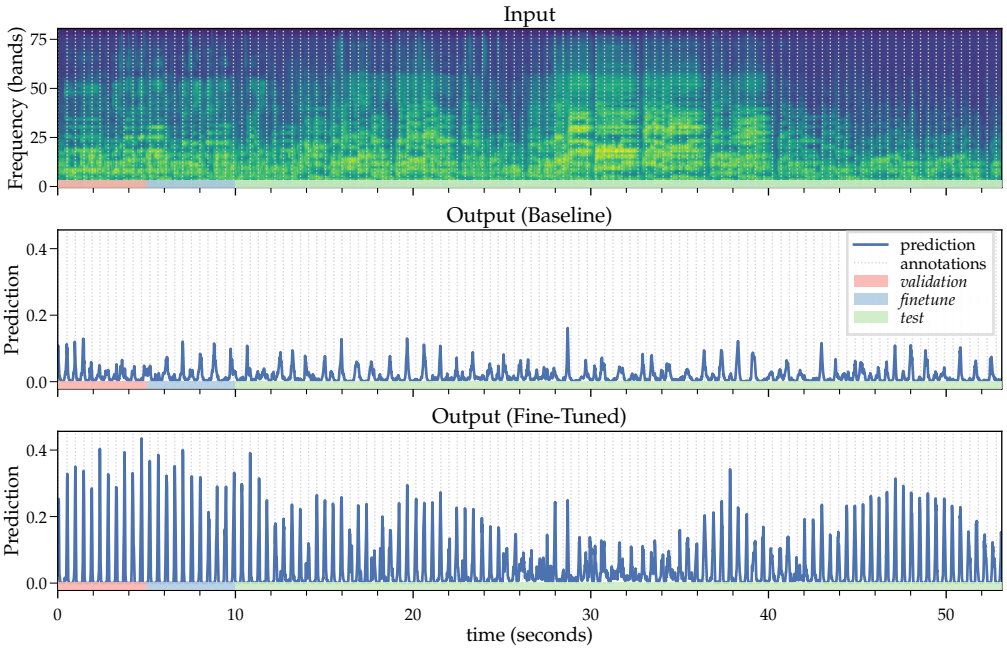


Figure 5. Network outputs for the baseline and fine-tuning approaches on *Blue Moon*. The *validation* region is composed by the 5 s after the first beat annotation (red), the *finetune* region by the following 5 s (blue) and the *test* region starting immediately after and going until the end of the file (green).

In terms of quantifying the improvement, we can see in Table 4 that when we fine-tuned, the number of required editing operations fell from eighty-three to eight, thus demonstrating the impact that a small number of annotations can have in transforming the efficacy of the baseline network for challenging content. To see this effect visually, we can plot precisely which operations are required and at which time instants both for the baseline and fine-tuned approach, as shown in Figure 6. In the upper plot of the figure, we can observe the high number of insertions, which is indicative of the baseline approach estimating a slower metrical level than the annotations. While it is possible to interpolate a set of beat detections to twice the tempo, this is only straightforward in cases where the tempo is largely constant. From the regions around 8 s–11 s and likewise from 25 s–32 s, there are numerous shift operations as well, indicating that the HMM was not able to make reliable beat detections in this region. By contrast, we see far fewer operations in the lower plot with the fine-tuned beat activation function, all of which are shifts in the form of minor timing corrections. Indeed, close inspection of the region right at the end of the excerpt (beyond the 50 s mark) highlights an interesting facet that the peaks of the beat activation function are strong, but misaligned with the annotations. Listening back to the manual annotations and the source audio, we could confirm that these specific annotations were drifting out of phase and should be corrected.

Table 4. Annotation efficiency (ae), correct detections (#det) and insertions (#ins), deletions (#del), shifts (#shf) and total edit operations (#ops) for *Blue Moon*.

	ae	**#det**	**#ins**	**#del**	**#shf**	**#ops**
Baseline	0.272	31	56	0	27	83
Fine-Tuned	0.930	107	0	1	7	8

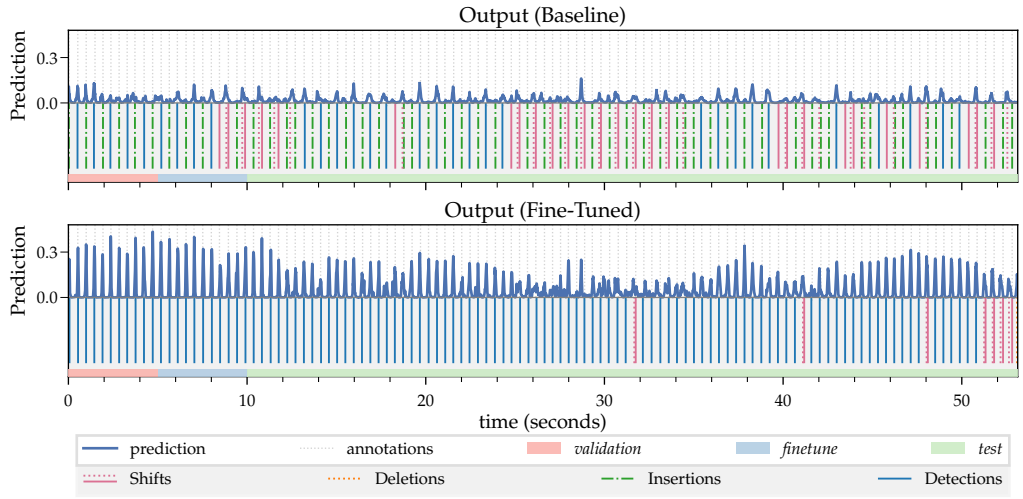


Figure 6. Network outputs for the baseline and fine-tuning approaches on *Blue Moon*. The *validation* region is composed by the 5 s after the first beat annotation (red), the *finetune* region by the following 5 s (blue) and the *test* region starting immediately after and going until the end of the excerpt (green). The dark blue solid line indicates the network prediction. The vertical grey dotted lines show the ground truth annotations. The vertical light blue solid lines show the correct beat detections. The incorrect beat outputs are notated with the required operation colour (delete—orange, shift—pink, insert—green).

6.3.2. *Choros №1*

The *Blue Moon* example from the previous section was selected in part due to its challenging musical properties, but also since it could be identified as among the excerpts from the *Hainsworth* dataset whose F-measure score was most improved by fine-tuning. In this section, we move away from excerpts in existing annotated datasets and instead look towards a simulation of our real-world use case. For this example, we chose a highly expressive solo guitar performance of the Heitor Villa-Lobos composition *Choros №1* as performed by Kyuhee Park (for reference, the specific performance can be found at the following url: https://www.youtube.com/watch?v=Uj_OferFIMk (accessed 25 May 2021)). Rather than using a minute-long excerpt, we examined the piece in its full duration of 4 m 51 s. A particular characteristic of this piece and something that is especially prominent in this specific performance is the extreme use of *rubato*—a property that is challenging for musical audio beat-tracking systems since it diverges strongly from the notion of a regular pulse. Indeed, the ground truth annotation of this piece, conducted entirely by hand in Sonic Visualiser [58], was very time-consuming and required frequent reference to the score to resolve ambiguities.

In Figure 7, we show the score representation of the beginning of the piece, including the *anacrusis* and the first complete bar. The *anacrusis* is important as it represents the main *motif* of the piece, recurring in several locations across its duration. It is composed of three sixteenth notes with *fermata*, indicating that the notes should be prolonged beyond the normal duration—at the discretion of the performer. This notation instructs the performer to an almost ad libitum interpretation, which results in extensive *rubato* across the full piece. Within the recording, these three sixteenth notes are clearly sounded by plucking, and given the absence of other instruments, they would be straightforward to detect even for a naive energy-based onset detection scheme. However, in the recording, they last over 4 s in duration and are thus highly problematic for beat tracking, because by reference to the score, all three occur within one notated beat.

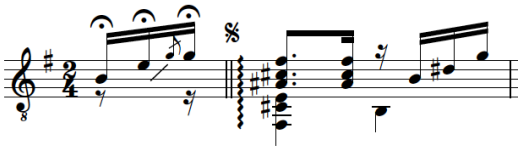


Figure 7. Excerpt of the *Choros №1* score (until the end of the first complete bar).

Since the analysis of this piece is not within the domain of annotated datasets, we adapted our fine-tuning strategy and expanded the region for fine-tuning to cover the first 15 s of the piece without validation and used the maximum number of epochs. Besides this alteration, we left all other aspects of the fine-tuning process described in Section 4 identical.

In the plots in Figure 8, the occurrences of this musical phrase are clearly depicted by a pattern in the log magnitude spectrogram input of the network in conjunction with the absence of beat annotations. The beat activation function of the baseline network output shows a strong indication of beats at these locations, whereas when performing fine-tuning, the beat activation is close to zero across all occurrences of the *motif*, despite the existence of clear onsets. In contrast to the *Blue Moon* example in which we observed the network adapt to a specific kind of spectrotimbral pattern to convey the beat, here we find evidence that the fine-tuning process has allowed the network to learn what is **not** the beat.

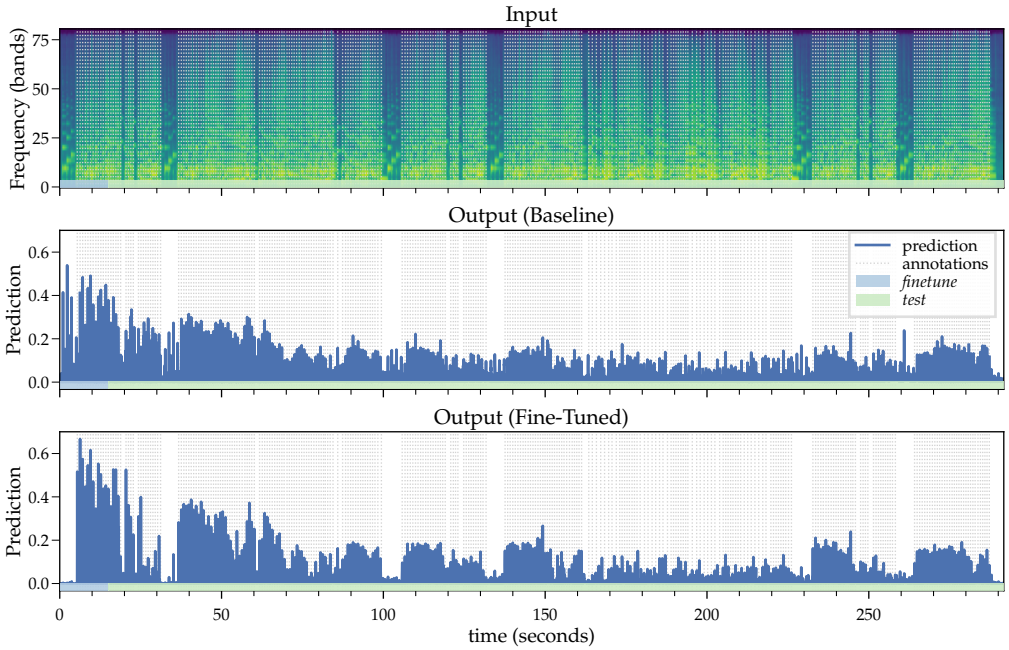


Figure 8. Network input and outputs for the baseline and fine-tuning approaches on *Choros №1*. *Finetune* region 0–15 s (blue) and the *test* region starting at 15 s (green).

The adaptation produced by the fine-tuning process has a clear impact from a practical point of view, as shown in Figure 9 and Table 5, with fewer editing operations required. From the zoomed in plot in Figure 9, we can see how well the fine-tuned network learned to ignore the motif once it occurred again just after the 30 s point. Indeed, here we observe a potential downside of the normally advantageous property of the HMM to fill gaps in a plausible way, as we see spurious detections from the fine-tuned network, which must be deleted. This behaviour, while specific to this piece, indicates that for highly expressive music including pulse suspensions, it may be worthwhile to consider a piecewise use of the HMM to prevent these gaps from being filled, e.g., based on the manual selection of temporal regions for inference, or in an automatic way by segmenting and excluding so-called "no beat" regions, as in [59].

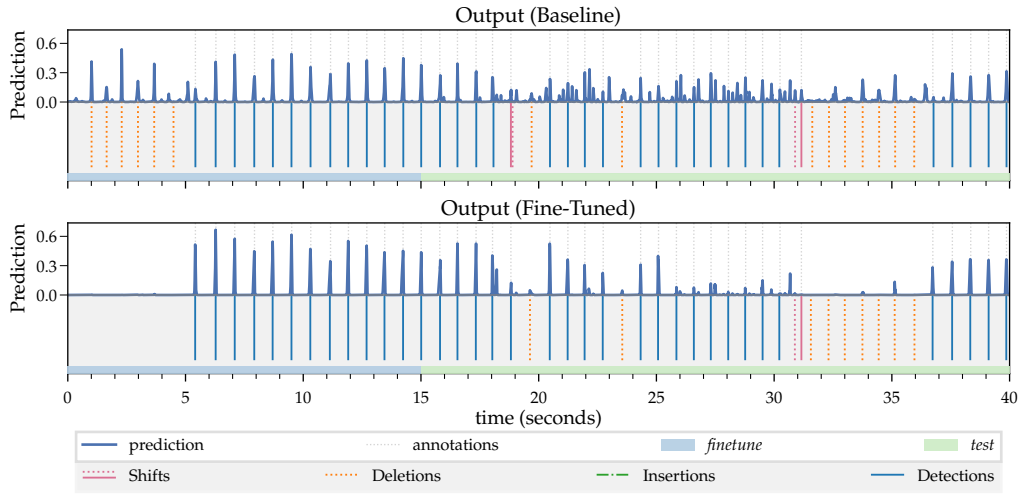


Figure 9. Network outputs for the baseline and fine-tuning approaches on *Choros №1* (zoomed over the initial 40 s). *Finetune* region 0–15 s (blue) and the *test* region starting at 15 s (green). The dark blue solid line indicates the network prediction. The vertical grey dotted lines show the ground truth annotations. The vertical light blue solid lines show the correct beat detections. The incorrect beat outputs are noted with the required operation colour (delete—orange, shift—pink, insert—green) to correct the annotation.

Table 5. Annotation efficiency (ae), correct detections (#det) and insertions (#ins), deletions (#del), shifts (#shf) and total edit operations (#ops) for *Choros №1*.

	ae	#det	#ins	#del	#shf	#ops
Baseline	0.555	207	0	69	97	166
Fine-Tuned	0.654	236	0	57	68	125

6.3.3. Catastrophic Forgetting

In the final part of our evaluation, we considered the impact of fine-tuning from a different perspective. Having established that fine-tuning is beneficial at the level of individual pieces, we now re-assess the performance of a fine-tuned network adapted to a given piece on other data. To this end, we investigated the presence and extent of "catastrophic forgetting." Known also as catastrophic interference, catastrophic forgetting is a well-known problem for backpropagation-based models [60] and is characterized by the tendency of an artificial neural network to abruptly forget previously learned information upon learning new information. Despite the sequential learning nature of our fine-tuning adaptation, this is merely episodic, as opposed to the continual acquisition of incrementally available information, which is more commonly addressed in catastrophic interference [61]. Nevertheless, it is of interest in the context of this work to examine what a fine-tuned network loses in terms of general knowledge about the beat when adapted to the properties of a specific piece of music.

To explore this behaviour, we return to the *Blue Moon* excerpt from the *Hainsworth* dataset. Across the training epochs of this excerpt, we evaluated the performance of each of the corresponding 24 models over the *GTZAN* and *TapCorrect* datasets. More specifically, for every epoch of the fine-tuning of *Blue Moon*, we saved the intermediate network and used it to estimate the beat in every excerpt of the *GTZAN* and *TapCorrect* datasets. In this way, we repeated the evaluation over these datasets 24 separate times.

Thus far, we have shown that, for this piece, there is a dramatic improvement in the F-measure once the fine-tuning has completed. However, we have not observed the manner in which the F-measure improves over the intermediate training epochs, nor how the fine-tuning process (i.e., specific to this musical excerpt) impacts performance on other

musical content. In the presence of catastrophic forgetting, we should expect some kind of inverse relationship in performance, with the improvement on *Blue Moon* coming at the expense of that on *GTZAN* and *TapCorrect*. In Figure 10, we plot this relationship over 24 epochs and indicate that early stopping occurs at Epoch 18.

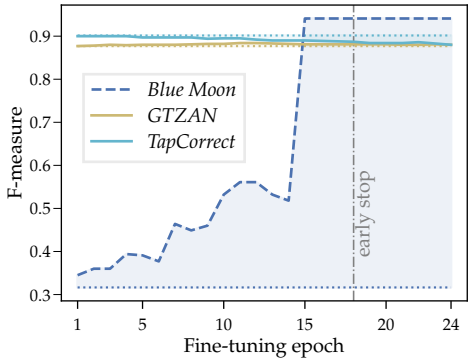


Figure 10. Evolution of F-measure during fine-tuning of *Blue Moon* on the *GTZAN* and *TapCorrect* datasets. Solid lines correspond to the fine-tuned model and dotted lines to the baseline model.

From the inspection of Figure 10, we can observe a rather nonlinear, and indeed nonmonotonic, increase in performance for *Blue Moon*. Between Epochs 15 and 16, there is a sudden jump in performance, after which the F-measure saturates above 0.90. Looking at the performance across the annotated datasets, we can see that the performance for *GTZAN* is essentially unchanged, and for *TapCorrect*, the F-measure falls by fewer than three percentage points. While our analysis was limited to fine-tuning on a single excerpt, it would appear that there was a very limited drop in performance due to the adaption of the network to *Blue Moon*. Indeed, if we considered that there were approximately 116 k weights in the baseline model and that we gave the network a very small temporal observation of 5 s, to which the network adapted with a reduced learning rate (one-fifth of the baseline training), we should perhaps not be surprised that a great proportion of the network weights remained unchanged. At this stage, we leave deeper analysis of this aspect as a topic for future work.

7. Discussion and Conclusions

In this paper, we explored the use of excerpt-specific fine-tuning of a state-of-the-art beat tracking system based on exposure to a very small annotated region. Across existing datasets, we demonstrated that this approach can lead to improved performance over the state of the art, and furthermore, we illustrated its potential to adapt to challenging conditions in terms of timbre and musical expression. We believe that the principal contribution of this work was to demonstrate the potential of fine-tuning within a user-driven annotation workflow and thus to provide a path towards very accurate analysis on highly challenging musical pieces. Within the wider context of beat tracking, we foresee that this type of approach could be used as a means for rapid, semi-automatic annotation of musical pieces to expand the amount of challenging annotated data for training new approaches. To this end, we will pursue the integration of our fine-tuning approach within a dedicated user interface for annotation, e.g., Sonic Visualiser [58].

In spite of the promising results obtained, it is important to recognise several limitations of our work and how they may be addressed in the future. First, our comparison against the state of the art was arguably tilted in favour of the fine-tuned approach, since per excerpt, we essentially created a new model and compared it to a single general model trained over a large amount of data. That said, our evaluation was carefully designed to exclude the interaction of the trained part of the input signal at inference, and furthermore, we did not claim that our fine-tuned approach represents a new state of the art. We simply sought to demonstrate that fine-tuning can be successfully applied across a large amount

and variety of musical material. Second, our evaluation was dependent on a rather arbitrary selection of two 5 s regions for training and validation; of course, we can expect that as we increase the duration of these regions, then we will likely obtain better performance for the piece in question, but doing so would require increased annotation effort on the part of the user, which we sought to minimize as much as possible. Indeed, in the limit, this would resolve to the user annotating the entire piece without any need for an automated solution at all.

Concerning the location of these regions, this was largely dictated by the goal of providing a "fair" comparison with the baseline network. A specific limiting factor of this deterministic assignment of the training region is that if the musical content in the remainder of the piece differs greatly from the information available for fine-tuning, then we should not expect it to be beneficial. To this extent, we may be underestimating the performance of our approach.

Within a real-world context, we foresee two main differences: (i) the end-user could choose where to annotate and for what proportion of the piece; and (ii) it would likely be advantageous not to exclude the region that has been exposed to the network at the time of inference. Beyond the presence of sharp peaks in the beat activation function, the user-provided beat annotations could also be harnessed for a more content-specific parameterisation of the inference technique, e.g., by setting an appropriate tempo range or some other parameterisation targeted for the presence of expressive timing [31]. As such, we believe that the real validation of our approach is not rooted in existing annotated datasets, but in a future user study that investigates how this approach can aid the annotation workflow. At this stage, we considered such an evaluation premature and reliant on first establishing, in quantitative terms, that fine-tuning is viable. However, in the future, we intend to gain deeper insight into how this approach could be used for data annotation, as well as understanding the impact and effort of the different correction operations. At the moment, we treated insertions, deletions and shifts as if they were equal for the calculation of the annotation efficiency, but we recognise that this is a simplification.

From a technical perspective, our approach to fine-tuning could be advanced in several ways. In our current implementation, we diverted from common practice in transfer learning between different tasks, which typically freezes all but the very last network layers, and instead unfroze all layers. In particular, we believe this is beneficial when it comes to analysing music that is unfamiliar from a timbre perspective and thus requires the adaptation of layers closer to the musical signal. However, we contend that there is significant potential to explore more advanced strategies including discriminative fine-tuning and gradual unfreezing [50], as well input-dependent fine-tuning, which could automatically determine which layers to fine-tune per target instance [62]. When considering the training regime, we also intend to explore novel ways in which the network adaptation could observe the entire piece, e.g., via semi-supervised learning, and thus overcome the limitations associated with fine-tuning based only on a partial observation of the input. Finally, looking beyond the task of musical audio beat tracking, we hope that our proposed fine-tuning methodology could be applied within other annotation-intensive MIR tasks.

Author Contributions: Conceptualization, A.S.P., S.B., J.S.C. and M.E.P.D.; data curation, A.S.P., S.B. and M.E.P.D.; formal analysis, A.S.P.; funding acquisition, J.S.C. and M.E.P.D.; investigation, A.S.P.; methodology, A.S.P., S.B. and M.E.P.D.; project administration, A.S.P. and M.E.P.D.; resources, A.S.P. and J.S.C.; software, A.S.P. and S.B.; supervision, M.E.P.D.; validation, A.S.P. and M.E.P.D.; visualization, A.S.P.; writing—original draft, A.S.P. and M.E.P.D.; writing—review and editing, A.S.P., S.B., J.S.C. and M.E.P.D. All authors read and agreed to the published version of the manuscript.

Funding: António Sá Pinto is supported by the FCT—Foundation for Science and Technology, I.P.—under Grant SFRH/BD/120383/2016. This research was also supported by national funds through the FCT—Foundation for Science and Technology, I.P.—under the projects IF/01566/2015 and CISUC—UID/CEC/00326/2020 and by the European Social Fund, through the Regional Operational Program Centro 2020.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BLSTM	Bidirectional long short-term memory model
DBN	Dynamic Bayesian network
DNN	Deep neural network
HMM	Hidden Markov model
MIR	Music information retrieval
MIREX	Music Information Retrieval Evaluation eXchange
STFT	Short-time Fourier transform
TCN	Temporal convolutional network

References

1. Schlüter, J.; Böck, S. Improved musical onset detection with Convolutional Neural Networks. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 6979–6983. [CrossRef]
2. Schreiber, H.; Müller, M. Musical tempo and key estimation using convolutional neural networks with directional filters. In Proceedings of the Sound and Music Computing Conference (SMC), Malaga, Spain, 28–31 May 2019; pp. 47–54.
3. Cemgil, A.T.; Kappen, B. Monte Carlo Methods for Tempo Tracking and Rhythm Quantization. *J. Artif. Intell. Res.* **2003**, *18*, 45–81. [CrossRef]
4. Hainsworth, S. Beat Tracking and Musical Metre Analysis. In *Signal Processing Methods for Music Transcription*; Klapuri, A., Davy, M., Eds.; Springer US: Boston, MA, USA, 2006; pp. 101–129. [CrossRef]
5. Sethares, W.A. *Rhythm and Transforms*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.
6. Müller, M. Tempo and Beat Tracking. In *Fundamentals of Music Processing*; Springer International Publishing: Berlin/Heidelberg, Germany, 2015; pp. 303–353. [CrossRef]
7. Stark, A.M.; Plumbley, M.D. Performance Following: Real-Time Prediction of Musical Sequences Without a Score. *IEEE Trans. Audio Speech Lang. Process.* **2011**, *20*, 190–199. [CrossRef]
8. Nieto, O.; Mysore, G.J.; Wang, C.i.; Smith, J.B.L.; Schlüter, J.; Grill, T.; McFee, B. Audio-Based Music Structure Analysis: Current Trends, Open Challenges, and Applications. *Trans. Int. Soc. Music. Inf. Retr.* **2020**, *3*, 246–263. [CrossRef]
9. Fuentes, M.; Maia, L.S.; Rocamora, M.; Biscainho, L.W.; Crayencour, H.C.; Essid, S.; Bello, J.P. Tracking beats and microtiming in Afro-latin American music using conditional random fields and deep learning. In Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR), Delft, The Netherlands, 4–8 November 2019; pp. 251–258.
10. Vande Veire, L.; De Bie, T. From raw audio to a seamless mix: creating an automated DJ system for Drum and Bass. *EURASIP J. Audio Speech Music. Process.* **2018**, *2018*. [CrossRef]
11. Davies, M.E.P.; Hamel, P.; Yoshii, K.; Goto, M. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2014**, *22*, 1726–1737. [CrossRef]
12. Bello, J.; Duxbury, C.; Davies, M.; Sandler, M. On the Use of Phase and Energy for Musical Onset Detection in the Complex Domain. *IEEE Signal Process. Lett.* **2004**, *11*, 553–556. [CrossRef]
13. Dixon, S. Onset detection revisited. In Proceedings of the 9th International Conference on Digital Audio Effects (DAFx), Montreal, QC, Canada, 18–20 September 2006; pp. 133–137.
14. Davies, M.E.P.; Plumbley, M.D. Context-Dependent Beat Tracking of Musical Audio. *IEEE Trans. Audio Speech Lang. Process.* **2007**, *15*, 1009–1020. [CrossRef]
15. Klapuri, A.P.; Eronen, A.J.; Astola, J.T. Analysis of the meter of acoustic musical signals. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 342–355. [CrossRef]
16. Dixon, S. An Interactive Beat Tracking and Visualisation System. In Proceedings of the International Computer Music Conference (ICMC), Havana, Cuba, 17–22 September 2001; pp. 215–218.
17. Goto, M.; Muraoka, Y. A beat tracking system for acoustic signals of music. In *Proceedings of the 2nd ACM International Conference on Multimedia (MULTIMEDIA '94)*; ACM Press: New York, NY, USA, 1994; pp. 365–372. [CrossRef]
18. Ellis, D.P.W. Beat Tracking by Dynamic Programming. *J. New Music Res.* **2007**, *36*, 51–60. [CrossRef]
19. Böck, S.; Schedl, M. Enhanced beat tracking with context-aware neural networks. In Proceedings of the 14th International Conference on Digital Audio Effects (DAFx), Paris, France, 19–23 September 2011; pp. 135–139.
20. Böck, S.; Krebs, F.; Widmer, G. A Multi-model Approach to Beat Tracking Considering Heterogeneous Music Styles. In Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 603–608.
21. Krebs, F.; Sebastian, B.; Widmer, G. An Efficient State-Space Model for Joint Tempo and Meter Tracking. In Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR), Malaga, Spain, 26–30 October 2015; pp. 72–78.

22. Böck, S.; Davies, M.E.P. Deconstruct, Analyse, Reconstruct: How To Improve Tempo, Beat, and Downbeat Estimation. In Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR), Montreal, QC, Canada, 12–16 October 2020; pp. 574–582.
23. Hainsworth, S. Techniques for the Automated Analysis of Musical Audio. Ph.D. Thesis, University of Cambridge, Cambridge, UK, 2004.
24. Davies, M.E.P.; Degara, N.; Plumbley, M.D. *Evaluation Methods for Musical Audio Beat Tracking Algorithms*; Technical Report October; Queen Mary University of London: London, UK, 2009.
25. Marchand, U.; Peeters, G. Swing Ratio Estimation. In Proceedings of the 18th International Conference on Digital Audio Effects (DAFx), Trondheim, Norway, 30 November–1 December 2015; pp. 423–428.
26. 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 Conference (ISMIR), Curitiba, Brazil, 4–8 November 2013; pp. 227–232.
27. Peeters, G. The Deep Learning Revolution in MIR: The Pros and Cons, the Needs and the Challenges. In *Perception, Representations, Image, Sound, Music—Proceedings of the 14th International Symposium (CMMR 2019), Marseille, France, 14–18 October 2019*; Revised Selected Papers; Kronland-Martinet, R., Ystad, S., Aramaki, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12631, pp. 3–30. [CrossRef]
28. Holzapfel, A.; Davies, M.E.P.; Zapata, J.R.; Oliveira, J.L.; Gouyon, F. Selective sampling for beat tracking evaluation. *IEEE Trans. Audio Speech Lang. Process.* **2012**, *20*, 2539–2548. [CrossRef]
29. Grosche, P.; Müller, M.; Sapp, C.S. What Makes Beat Tracking Difficult? A Case Study on Chopin Mazurkas. In Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR), Utrecht, The Netherlands, 9–13 August 2010; pp. 649–654.
30. Dalton, B.; Johnson, D.; Tzanetakis, G. DAW-Integrated Beat Tracking for Music Production. In Proceedings of the Sound and Music Computing Conference (SMC), Malaga, Spain, 28–31 May 2019; pp. 7–11.
31. Pinto, A.S. Tapping Along to the Difficult Ones: Leveraging User-Input for Beat Tracking in Highly Expressive Musical Content. In *Perception, Representations, Image, Sound, Music—Proceedings of the 14th International Symposium, CMMR 2019, Marseille, France, 14–18 October 2019*; Revised Selected Papers; Kronland-Martinet, R., Ystad, S., Aramaki, M., Eds.; Springer: Berlin/Heidelberg, Germany, 2021; Volume 12631, pp. 75–90. [CrossRef]
32. Pons, J.; Serra, J.; Serra, X. Training Neural Audio Classifiers with Few Data. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 16–20. [CrossRef]
33. Pan, S.J.; Yang, Q. A survey on transfer learning. *IEEE Trans. Knowl. Data Eng.* **2010**, *22*, 1345–1359. [CrossRef]
34. van den Oord, A.; Dieleman, S.; Schrauwen, B. Transfer Learning by Supervised Pre-training for Audio-based Music Classification. In Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 29–34.
35. Choi, K.; Fazekas, G.; Sandler, M.; Cho, K. Transfer learning for music classification and regression tasks. In Proceedings of the 18th International Conference on Music Information Retrieval (ISMIR), Suzhou, China, 23–27 October 2017; pp. 141–149.
36. Burloiu, G. Adaptive Drum Machine Microtiming with Transfer Learning and RNNs. Extended Abstracts for the Late-Breaking Demo Session of the International Society for Music Information Retrieval Conference (ISMIR). 2020. Available online: https://program.ismir2020.net/static/lbd/ISMIR2020-LBD-422-abstract.pdf (accessed on 25 May 2021).
37. Fiocchi, D.; Buccoli, M.; Zanoni, M.; Antonacci, F.; Sarti, A. Beat Tracking using Recurrent Neural Network: A Transfer Learning Approach. In Proceedings of the 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; pp. 1915–1919. [CrossRef]
38. Wang, Y.; Yao, Q.; Kwok, J.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-Shot Learning. *arXiv* **2019**, arXiv:1904.05046.
39. Choi, J.; Lee, J.; Park, J.; Nam, J. Zero-shot learning for audio-based music classification and tagging. In Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR), Delft, The Netherlands, 4–8 November 2019; pp. 67–74.
40. Dhillon, G.S.; Chaudhari, P.; Ravichandran, A.; Soatto, S. A Baseline for Few-Shot Image Classification. In Proceedings of the 8th International Conference on Learning Representations (ICLR), Addis Ababa, Ethiopia, 26–30 April 2020.
41. Manilow, E.; Pardo, B. Bespoke Neural Networks for Score-Informed Source Separation. *arXiv* **2020**, arXiv:2009.13729.
42. Wang, Y.; Salamon, J.; Cartwright, M.; Bryan, N.J.; Bello, J.P. Few-Shot Drum Transcription in Polyphonic Music. In Proceedings of the 21st International Society for Music Information Retrieval Conference (ISMIR), Montreal, QC, Canada, 12–16 October 2020; pp. 117–124.
43. Davies, M.E.P.; Böck, S. Temporal convolutional networks for musical audio beat tracking. In Proceedings of the 27th European Signal Processing Conference (EUSIPCO), A Coruna, Spain, 2–6 September 2019.
44. Böck, S.; Davies, M.E.P.; Knees, P. Multi-Task Learning of Tempo and Beat: Learning One To Improve the Other. In Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR), Delft, The Netherlands, 4–8 November 2019; pp. 486–493.
45. Böck, S.; Krebs, F.; Widmer, G. Joint Beat and Downbeat tracking with recurrent neural networks. In Proceedings of the 17th International Society for Music Information Retrieval Conference (ISMIR), New York, NY, USA, 7–11 August 2016; pp. 255–261.

46. Gouyon, F.; Klapuri, A.; Dixon, S.; Alonso, M.; Tzanetakis, G.; Uhle, C.; Cano, P. An Experimental Comparison of Audio Tempo Induction Algorithms. *IEEE Trans. Audio Speech Lang. Process.* **2006**, *14*, 1832–1844. [CrossRef]
47. Hockman, J.A.; Bello, J.P.; Davies, M.E.P.; Plumbley, M.D. Automated Rhythmic Transformation of Musical Audio. In Proceedings of 11th International Conference on Digital Audio Effects (DAFx), Espoo, Finland, 1–4 September 2008; pp. 177–180.
48. Gouyon, F. A Computational Approach to Rhythm Description—Audio Features for the Computation of Rhythm Periodicity Functions and their use in Tempo Induction and Music Content Processing. Ph.D. Thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2005.
49. Yosinski, J.; Clune, J.; Bengio, Y.; Lipson, H. How transferable are features in deep neural networks? In Proceedings of the Advances in Neural Information Processing Systems (NIPS2014), Montreal, QC, Canada, 13 December 2014; Volume 27. [CrossRef]
50. Howard, J.; Ruder, S. Universal language model fine-tuning for text classification. In *ACL 2018—Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*; Association for Computational Linguistics Location: Melbourne, Australia, 2018; pp. 328–339. [CrossRef]
51. Pinto, A.S.; Domingues, I.; Davies, M.E.P. Shift If You Can: Counting and Visualising Correction Operations for Beat Tracking Evaluation. *arXiv* **2020**, arXiv:2011.01637.
52. Li, M.; Chen, X.; Li, X.; Ma, B.; Vitányi, P.M. The similarity metric. *IEEE Trans. Inf. Theory* **2004**, *50*, 3250–3264. [CrossRef]
53. Valero-Mas, J.J.; Iñesta, J.M. Interactive user correction of automatically detected onsets: approach and evaluation. *EURASIP J. Audio Speech Music Process.* **2017**, *2017*. [CrossRef]
54. Driedger, J.; Schreiber, H.; De Haas, W.B.; Müller, M. Towards automatically correcting tapped beat annotations for music recordings. In Proceedings of the 20th International Society for Music Information Retrieval Conference (ISMIR), Delft, The Netherlands, 4–8 November 2019; pp. 200–207.
55. Faisal, A.A.; Selen, L.P.; Wolpert, D.M. Noise in the nervous system. *Nat. Rev. Neurosci.* **2008**, *9*, 292–303. [CrossRef]
56. Raffel, C.; Mcfee, B.; Humphrey, E.J.; Salamon, J.; Nieto, O.; Liang, D.; Ellis, D.P.W. mir_eval: A Transparent Implementation of Common MIR Metrics. In Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR), Taipei, Taiwan, 27–31 October 2014; pp. 367–372.
57. Tzanetakis, G.; Cook, P. Musical genre classification of audio signals. *IEEE Trans. Speech Audio Process.* **2002**, *10*, 293–302. [CrossRef]
58. Cannam, C.; Landone, C.; Sandler, M. Sonic visualiser. In *Proceedings of the International Conference on Multimedia (MM '10)*; ACM Press: New York, NY, USA, 2010; pp. 1467–1468. [CrossRef]
59. Schreiber, H.; Müller, M. A single-step approach to musical tempo estimation using a convolutional neural network. In Proceedings of the 19th International Society for Music Information Retrieval Conference (ISMIR), Paris, France, 23–27 September, 2018; pp. 98–105.
60. McCloskey, M.; Cohen, N.J. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. *Psychol. Learn. Motiv. Adv. Res. Theory* **1989**, *24*, 109–165. [CrossRef]
61. Parisi, G.I.; Kemker, R.; Part, J.L.; Kanan, C.; Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Netw.* **2019**, *113*, 54–71. [CrossRef] [PubMed]
62. Guo, Y.; Shi, H.; Kumar, A.; Grauman, K.; Rosing, T.; Feris, R. SpotTune: Transfer Learning through Adaptive Fine-tuning. *arXiv* **2018**, arXiv:1811.08737.

HIGH FREQUENCY MAGNITUDE SPECTROGRAM RECONSTRUCTION FOR MUSIC MIXTURES USING CONVOLUTIONAL AUTOENCODERS

Marius Miron *

Independent researcher
miron.marius@gmail.com

Matthew E.P. Davies

INESC TEC
Sound and Music Computing Group
Porto, Portugal
mdavies@inesctec.pt

ABSTRACT

We present a new approach for audio bandwidth extension for music signals using convolutional neural networks (CNNs). Inspired by the concept of inpainting from the field of image processing, we seek to reconstruct the high-frequency region (*i.e.*, above a cutoff frequency) of a time-frequency representation given the observation of a band-limited version. We then invert this reconstructed time-frequency representation using the phase information from the band-limited input to provide an enhanced musical output. We contrast the performance of two musically adapted CNN architectures which are trained separately using the STFT and the invertible CQT. Through our evaluation, we demonstrate that the CQT, with its logarithmic frequency spacing, provides better reconstruction performance as measured by the signal to distortion ratio.

1. INTRODUCTION

Audio signals are often low-passed, encoded or compressed before transmitting them through phone lines and Internet streams. This results in the loss of high frequency content and compromises audio quality. Narrow-band audio signals which have information up to a certain frequency cutoff can be perceptually enhanced by reconstructing the higher frequency content. This research task, known as *audio bandwidth extension*, attempts to increase the perceived or real frequency spectrum of audio signals [1, 2, 3, 4, 5].

Audio bandwidth extension methods have been applied to speech signals in an unsupervised and supervised manner. The former are typically statistical approaches which model the relationship between low and high frequency spectral content by relating lower and upper harmonics [1]. For instance, the linear predictive coding (LPC) method in [2] analyzes the lower frequency spectra to synthesize high frequency components. It relies on a codebook: a dictionary of wide-band envelopes, which are matched with the envelope of narrow-band spectral frames. Spectral band replication [6] on the other hand transposes up harmonics from lower and midrange frequencies to higher bands.

Supervised methods learn priors from wide-band signals which are later used to recover the high frequency content of narrow-band signals. Matrix decomposition methods such as non-negative matrix factorization (NMF) [3, 5] treat the magnitude spectrogram as combinations of priors in the form of non-negative bases. At the test stage, these bases are kept fixed and are used to estimate the NMF parameters which best explain the narrow-band signal.

Methods using neural networks learn priors from features derived from time-frequency representations to predict high-band spectral envelopes [7, 8]. Bandwidth extension with deep neural networks has been shown to increase the robustness of speech recognition [8]. In addition, the resolution of raw audio signals, regarded as time series, can be increased using convolutional neural networks (CNNs) [9].

In this paper we seek to estimate high frequency components in time-frequency representations of music signals. Compared to speech, music signals are often complex mixtures, comprising a variety of instruments, both percussive and harmonic, singing voice, and non-linear audio effects. Thus, music signals have broader, richer, and perceptually more relevant high frequency content, which is therefore more difficult to estimate.

While the aim of bandwidth extension for speech is tightly coupled with signal compression and band-limited communication channels, for music signals there are important distinctions both in terms of the constraints of the problem and the potential applications. First and foremost, our aim is to perform bandwidth extension up to CD quality (*i.e.*, 44.1 kHz sampling rate with a Nyquist rate of 22.05 kHz). Given the absence of harmonic information in high frequency musical content (*e.g.*, above 10 kHz), our proposed musical bandwidth extension will be required to reconstruct percussive-type content. Depending on the bandwidth of the narrow-band input signal, it may also be required to reconstruct the upper partials of harmonic content present in the narrow-band signal. In this way, perceptually accurate musical bandwidth extension could be used to replace high-band information typically lost via lossy compression in audio formats such as MP3 and AAC, and thus reduce the bandwidth overhead when streaming music, or allocate a higher bit rate for lower frequency information.

Our specific long term goal is to explore a more creative application of audio bandwidth extension, namely towards the restoration of old music recordings. To this end, we seek to renew old recordings (in particular, jazz from the 1940s and 50s) and thus allow modern-day listeners to experience this music in high audio quality as performed by the original musicians. Towards this ambitious goal, we first investigate the feasibility of full-bandwidth extension for music signals under more controlled conditions, which can be more readily evaluated via access to both the full- and band-limited versions.

Similar to the concept of image inpainting or completion [10, 11], for which CNNs have been shown to be particularly adept, we aim to learn localized features in order to recover the missing higher frequency regions of short-term Fourier transform (STFT) and constant-Q transform (CQT) stereo magnitude spectrograms [12]. However, since the time and frequency axes in STFT and CQT representations do not correlate in the same way

* Marius Miron is currently a post-doctoral researcher at the European Commission Joint Research Center

A Perceptually-Motivated Harmonic Compatibility Method for Music Mixing

Gilberto Bernardes[1] Matthew E. P. Davies[1], and Carlos Guedes[1,2]

[1] Sound and Music Computing, INESC TEC
[2] New York University Abu Dhabi
{gba, mdavies} @inesctec.pt, carlos.guedes@nyu.edu

Abstract. We present a method for assisting users in the process of music mashup creation. Our main contribution is a harmonic compatibility metric between musical audio samples which combines existing perceptual relatedness (i.e., chroma vectors or key affinity) and consonance approaches. Our harmonic compatibility metric is derived from Tonal Interval Space indicators, which we adapt to robustly describe the harmonic content of musical audio. Additional attributes of key, rhythmic (density), and spectral content are computed from musical audio to enhance the compatibility representation of a sample collection in an interactive visualization. An evaluation of our harmonic compatibility method shows that it complies with the principles embodied in Western tonal harmony to a greater extent than previous approaches.

Keywords: music mashup; digital DJ interfaces; audio content analysis; music information retrieval.

1 Introduction

Mashup creation is a music composition practice strongly linked to the various sub-genres of Electronic Dance Music (EDM) and the role of the DJ [22]. It entails the recombination of existing (pre-recorded) musical audio as a means for creative endeavor [22]. As such, it can be seen as a byproduct of the existing mass preservation mechanisms and inscribed within the artistic view of the database as a symbol of postmodern culture [15]. Urged by the need to retrieve and manipulate musical audio from the ever-growing banks of digital music, mashup creation is typically confined to technology-fluent composers, as it requires expertise which extends from the understanding of musical structure to the navigation and retrieval of musical audio from large databases. Both industry and academia have been devoting efforts to enhance the experience of digital tools for mashup creation by streamlining the time-consuming search for compatible musical audio.

Early research on computational mashup creation, focused on rhythmic-only features, particularly those relevant to the temporal alignment of two or more musical tracks [10]. Recent research on this topic has expanded the range of musical attributes under consideration, notably including harmonic-driven features

to identify compatible musical audio, commonly referred to as *harmonic mixing*. We can identify three major harmonic mixing methods: key affinity, chroma vectors similarity, and (minimal) psychoacoustic dissonance. The affinity between musical keys is a prominent method in commercial applications which favors relative major-minor and intervals of fifth relations across musical keys [16]. Chroma vector similarity inspects the (cosine) distance between chroma vectors representations of pitch shifted versions of two given audio samples as a measure of their compatibility [6, 7, 14]. Psychoacoustic dissonance models have been used to search for pitch shifted versions of overlapping musical audio which minimize their combined level of sensory roughness [9].

While these approaches have shown to correlate well with user enjoyment, we argue that they misrepresent some aspects of harmonic compatibility. First, all possible overlaps between in-key (or diatonic) pitch configurations result in highly contrasting sonorities with significant levels of enjoyment [2], thus motivating a harmonic compatibility metric below the key level. Second, while chroma vector distances are effective in capturing highly similar matches between any two given audio samples, they lack a perceptually-aware basis for comparing pitch configurations [1], and can thus fail to provide an effective ranking between musical audio collections. Third, while psychoacoustic models show enhanced performance over remaining approaches, they not only prove to be of limited use when the spectral content of the samples do not overlap (or no interaction exists within each critical band), but also violate the harmonic principles embodied in Western music [12].

In light of these limitations, we propose a new method for computing the harmonic compatibility between a beat-matched collection of audio samples based on two indicators from the perceptually-motivated Tonal Interval Space: perceptual relatedness and consonance [1]. The proposed method, not only synthesizes the two latter approaches on a perceptually-relevant space, but also provides in-key harmonic compatibility metrics, irrespective of the spectral region each file occupies. Moreover, our method considers two additional dimensions that can help users defining compositional goals in terms of rhythmic (onset density) and spectral (region) content. A prototype created in Pure Data [19] provides an interactive visualization of the compatibility attributes, which allow users to intuitivly navigate through musical audio collections.

Fig. 1 shows the component modules of our harmonic mixing method which will be detailed in the remainder of this paper as follows. Sec. 2 reviews the Tonal Interval Space, which we adapt towards an enhanced representation of the harmonic content of musical audio. Sec. 3 presents content-driven harmonic, rhythmic, and spectral analysis of musical audio. Sec. 4 introduces a novel metric for computing the harmonic compatibility between audio samples. Sec. 5 details an interactive visualization which exposes the compatibility of an audio sample collection. Sec. 6 presents an evaluation of the harmonic compatibility indicator, which underpins the harmonic mixing method. Finally, Sec. 7 presents conclusions and areas for future work.

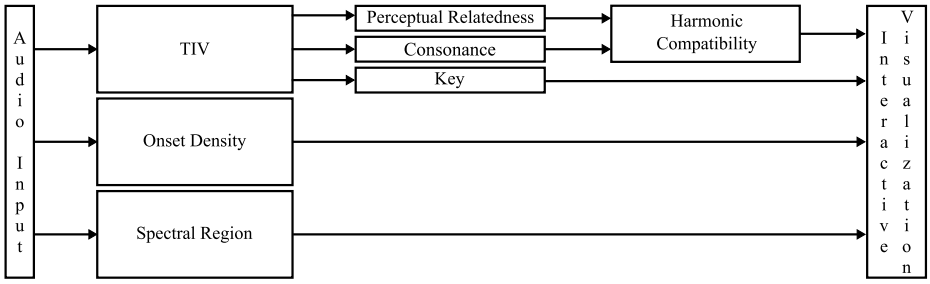


Fig. 1. Diagram of the component modules of our compatibility method for music mixing. Arrows indicate the data flux.

2 Adapting Tonal Interval Vectors for Musical Audio

We represent the harmonic content of musical audio samples as 12-dimensional Tonal Interval Vectors (TIVs) [1]. This vector space creates an extended representation of tonal pitch in the context of the *Tonnetz* [8], named Tonal Interval Space, where the most salient pitch levels of tonal Western music—pitches, chords, and keys—exist as unique locations. TIVs, $T(k)$, are computed from an audio signal as the weighted Discrete Fourier Transform (DFT) of a L_1 normalized chroma vector, $c(n)$, such that:

$$T(k) = w_a(k) \sum_{n=0}^{N-1} \bar{c}(n) e^{\frac{-j2\pi kn}{N}}, \quad k \in \mathbb{Z} \quad . \tag{1}$$

where $N = 12$ is the dimension of the chroma vector, each of which expresses the energy of the 12 pitch classes, and $w_a(k)$ are weights derived from empirical ratings of dyads consonance used to adjust the contribution of each dimension k (or interpreted musical interval) of the space, which we detail over the next paragraphs at length. We set k to $1 \leq k \leq 6$ for $T(k)$ since the remaining coefficients are symmetric. $T(k)$ uses $\bar{c}(n)$ which is $c(n)$ normalized by the DC component $T(0) = \sum_{n=0}^{N-1} c(n)$ to allow the representation and comparison of music at different hierarchical levels of tonal pitch [1]. To represent variable-length audio samples, we accumulate chroma vectors, $c(n)$, resulting from $\approx$ $372ms$ analysis windows with 50% overlap across the sample duration.

Table 1. Composite consonance ratings of dyads consonance [11].

Interval Class	m2/M7	M2/m7	m3/M6	M3/m6	P4/P5	TT
Consonance	-1.428	-.582	.594	.386	1.240	-.453

In [1], we used two complementary sources of empirical data—empirical ratings of dyads consonance shown in Table 1 [11] and the ranking order of tri-

ads consonance: {maj, min, sus4, dim, aug} [5]—to make the Tonal Interval Space perceptually relevant for symbolic music input representations (i.e., binary chroma vectors). Here, we revisit the task to comply with the timbral components of musical audio. Our goal is to find a set of weights, $w_a(k)$, which regulate the importance of the DFT coefficients k in Eq. 1, so that the space conveys a reliable consonance indicator correlated with the aforementioned empirical ratings of dyad [11] and triad consonance [5]. The applied method follows the previously used brute force approach [1], which guaranteed a near optimal result.

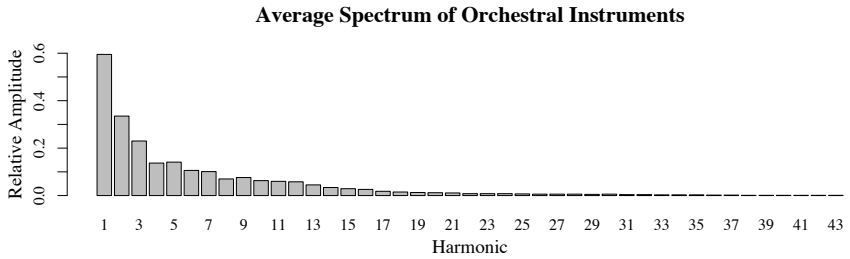


Fig. 2. Average harmonic spectrum of 1338 tones from orchestral instruments [18].

A major problem in defining a set of weights for robustly representing musical audio in the Tonal Interval Space is the variability of timbre across musical instruments and registers. A refined model capable of tracing the idiosyncratic timbral attributes of a particular instrument raises scalability and complexity issues which would defeat the value of the Tonal Interval Space in providing effective and, most importantly, efficient perceptual indicators of tonal pitch. To circumvent these issues, we adopt the 43-partials harmonic spectrum template shown in Fig. 2 to represent the harmonic content of musical audio. The template results from averaging 1338 recorded instrument tones from 23 Western orchestral instruments and can be understood as a time-invariant spectrum of an "average instrument" [18].

To allow a computationally tractable search for weights to represent musical audio in the Tonal Interval Space, we split the task into three steps. In the first step, we find the weights, $w_a(k)$, from all possible 6-element combinations (with repetition and order relevance), of the set $I = \{1, 19\} \in \mathbb{Z} : I = 2I + 1$ (a total of one million combinations), which maintain in the Tonal Interval Space the empirical ranking order of common triads consonance [5]. Following [1], we compute the consonance of musical audio triads in the Tonal Interval Space as the norm of TIVs, $\|T(k)\|$, which we fully detail in Sec. 3.2. In the second step, from the resulting set of 111 weight vectors which preserve the ranking order of empirical triads consonance (i.e., a Spearman rank correlation $\rho = 1$), we identify those which have the highest linear correlation to the empirical dyads consonance ratings shown in Table 1. In the third step, we perform a refined optimization

on the two weight vectors ($\{1, 7, 15, 11, 13, 7\}$ and $\{3, 7, 15, 11, 13, 7\}$) with the highest linear correlation ($r = .988$), below our minimal interval. To this end, we repeated steps 1 and 2 on 4096 6-element combinations (with repetition and order relevance) of the $\{0, .5, 1, 1.5\}$ set, which we add to the two aforementioned weight vectors.

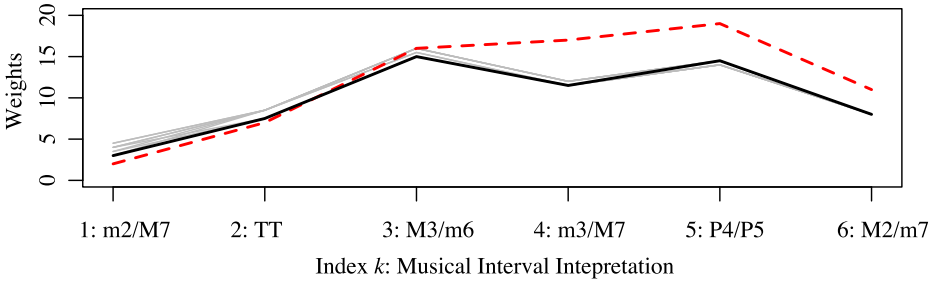


Fig. 3. The set of weights that maximize the linear ($r > 0.995$) and ranking order ($\rho = 1$) correlation of Tonal Interval Space's musical audio consonance indicator with empirical ratings of dyads and triads consonance, respectively. The bold line corresponds to the set of weights $w_a(k)$ used in Eq. 1 and the dashed line to the weights, $w_s(k)$, defined in [1] for a symbolic based Tonal Interval Space.

Fig. 3 shows 11 sets of weights, $w_a(k)$, which preserve empirical triads ($\rho = 1$) and dyads ($r > .99$) consonance for musical audio. Given the inherent similarity in shape of the different sets of weights and their almost perfect linear relationship, we do not believe the choice over exactly which set of weights to be critical. However, we ultimately selected the weights with the greatest mutual separation between the triads according to consonance, thus $w_a(k) = \{3, 8, 11.5, 11.5, 15, 14.5, 7.5\}$.

3 Musical Audio Analysis

3.1 Dissonance and Perceptual Relatedness

To provide a mathematical representation of Western tonal harmony perception as distances in the Tonal Interval Space, we distorted a DFT space according to the weights, $w_a(k)$, derived from empirical consonance ratings. In light of this design feature and following previous metrics detailed in [1], we can compute two indicators of consonance, C, and perceptual relatedness, R, from the space as distance metrics.

$$C = \|T(k)\| \quad , \tag{2}$$

$$C_{ij} = \frac{\|a_i T_i(k) \ + a_j T_j(k)\|}{a_i + a_j \sum_{k=1}^{M=6} w(k)} \quad , \tag{3}$$

and

$$R_{i,j} = \sqrt{\sum_{k=1}^{M} |T_i(k) - T_j(k)|^2} \quad . \tag{4}$$

Eq. 2 computes the norm of a TIV, $T(k)$, which we adopt as a measure of consonance of musical audio. Given that the location of multi-pitch TIVs, is equal to the linear combination of its component pitch classes [1], we can efficiently compute the consonance of two combined TIVs, $T_i(k)$ and $T_j(k)$, representing two overlapping audio samples, using Eq. 3, where a_i and a_j are the amplitudes of $T_i(k)$ and $T_j(k)$. Eq. 4 computes the perceptual relatedness, $R_{i,j}$, as the Euclidean distance between the TIVs, $T_i(k)$ and $T_j(k)$.

3.2 Musical Key

Using the method from [3], we estimate the global key of a musical audio sample, Q, in the Tonal Interval Space by finding the closest key TIV from an input audio TIV, as the minimum Euclidean distance function of an audio input TIV, $T(k)$, from the 12 major and 12 minor key TIVs, $T_p(k)$, such that:

$$Q = \text{argmin}_p \sqrt{\sum_{k=1}^{6} |T(k) \cdot \alpha - T_p(k)|^2} \quad . \tag{5}$$

where $T_p(k)$ is derived from a collection of templates (understood here as chroma vectors) representing pitch class distributions for each of the 12 major and 12 minor keys [21]. When $p \leqslant 11$, we adopt the major profile and when $p \geqslant 12$, the minor profile. $\alpha = 0.35$ is a factor which displaces input sample TIVs to balance predictions across modes [3]. The estimated key, Q, ranges between $0 - 11$ for major keys and $12 - 23$ for minor keys, where 0 corresponds to C major, 1 to C# major, and so on through to 23 being B minor.

3.3 Note Onset Density

We compute a note onset density function, D, from the musical audio input as a relevant rhythmic attribute to guide users in the search for samples from a collection which meet particular compositional goals. The task is performed by a threefold approach. First, we compute a spectral flux onset detection function, from a windowed power spectrum representation of the audio signal (window size $\approx$ 46 ms with 50% overlap), using the `timbreID` [4] library within Pure Data. Second, we extract the peaks from the function above a user-defined threshold, t, whose temporal location we assume to indicate note onset times. Prior to the peak detection stage, we apply a bi-directional low-pass IIR filter, with a cutoff frequency of $5Hz$ to avoid spurious detections. The note onset density, D, of an audio sample is then computed as the ratio between the number of onsets and the duration of the audio file in seconds.

3.4 Spectral Region

We use the perceptually motivated Bark frequency scale [23] to represent the spectral content of an input audio signal. From accumulated Bark spectrum representations across an audio file, we extract the centroid as an indicator of its spectral region, S, using Eq. 6. To compute the Bark spectrum, B_i, we use the `timbreID` [4] library within Pure Data, which warps a power spectrum to the 24 critical bands of the human auditory system (i.e., Bark bands). In adopting a Bark spectral representation, we balance the resolution across the human hearing range, with increased resolution in the low frequency region.

$$S = \frac{\sum_{i=1}^{19} B_i \cdot i}{\sum_{i=1}^{19} B_i} \quad . \tag{6}$$

where B_i is the energy of the bark band i. The S indicator can range from 1 to 24 Bark bands.

4 Harmonic Compatibility Measure

The level of harmonic compatibility is expressed as the combination of two harmonic audio indicators from the Tonal Interval Space detailed in Sec. 3.1: perceptual relatedness, R, and consonance, C. The first indicator finds sonorities which have a strong perceptual affinity and thus range from a perfect match to sonorities with different timbres and similar pitch content, to an array of sonorities with increased levels of perceptual distance. We envisage it as an extension of the chroma vector similarity method used as a measure of harmonic compatibility in prior studies [6, 7, 14].

By assuming that the prevalence of musical elements in tonal Western music primarily depends on their consonance and, to account for cases in which the same-level of perceptual relatedness, R, is exhibited, we merged it with the consonance indicator, C, to ensure some preference over more consonance resulting audio mixes, and disambiguate equally distant perceptual configurations. The resulting harmonic compatibility, H, between two samples, i and j, is then computed as the product of the two indicators:

$$H_{i,j} = \bar{R}_{i,j} \cdot (1 - \bar{C}_{ij}) \quad , \tag{7}$$

where $\bar{R}$ and $\bar{C}$ are R and C scaled to the range $\{0, 1\} \in \mathbb{R}$ to balance the importance of both indicators in the compatibility metric. In Eq. 7, we take $(1 - \bar{C})$ to match the $\bar{R}$ metric, where *small* distances values indicate *related* configurations.

5 Interactive Visualization

Inspired by [13], we created a prototype in Pure Data which provides an interactive visualization of the musical audio analysis detailed in Sec. 3 and 4

towards an enhanced search for compatible samples from a musical audio collection. Audio samples are represented as regular polygons (see Fig. 4), whose distance indicate their level of harmonic compatibility, H. The computation of coordinates for each musical audio sample from a square matrix of all pairwise samples distances (i.e., harmonic compatibility) is a classical problem, which can be solved by a specific class of algorithms, notably including Multidimensional Scaling (MDS). In greater detail, from m musical audio samples in a collection, we compute a $H_m \cdot H_m$ harmonic compatibility square matrix, from which an MDS representation extracts two-dimensional coordinates for each sample. The resulting representation attempts to preserve the inter-sample harmonic compatibility with minimal distortion.

Additional musical attributes of each sample are represented by graphical attributes of the polygons. The number of sides, ranging from three to six, expose the note onset density, D. The mapping between the note onset density, D, and the polygons' sides is done by scaling and rounding the D values to the range $\{3, 6\} \in \mathbb{Z}$. The higher the number of onsets, the higher the number of sides of the polygon. The spectral region of each sample is represented by the polygons' color, ranging from continuous shades of yellow to red. The mapping between the color scheme and the spectral region (i.e., the centroid of the bark spectrum representation for 24 bands) is linear. Bark band 1 corresponds to yellow, and bark band 24 to red. Between these values, the colors are linearly mixed.

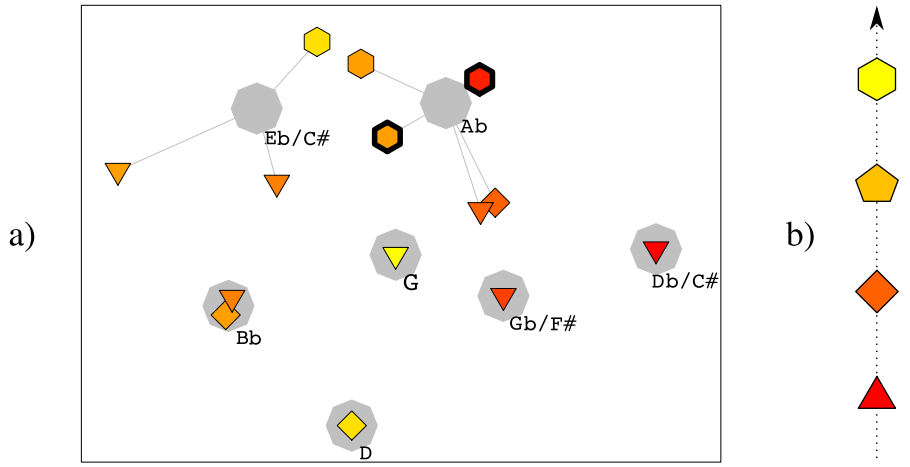


Fig. 4. a) Interactive visualization of the harmonic compatibility (distance) between musical audio samples from a collection. The graphical attributes of the polygons, representing individual samples, show onset density (number of sides) and spectral region (color). Polygons with thick lines indicate selected files currently playing and the octagons key locations. b) Ranking order of low to high onset density and spectral region.

The user can interact with the visualization by clicking on the polygons to trigger their playback, thus promoting an intuitive search for compatible samples as well as strategies for serendipity and experimentation, rather than a fully automatic method for mashup creation. Users can trigger up to 4 samples simultaneously, whose durations are then time-stretched (while preserving the original pitch content) to the minimal duration of the active files for synchronization. A demo of this interactive visualization can be found online at: `http://bit.ly/2pGvAnO`.

6 Evaluation

To evaluate our harmonic mixing method, we address three aspects of the system which underpin the search for harmonically compatible musical samples at the chordal level—at which most mashup creation exists. First, we demonstrate how the weights, $w_a(k)$, implemented as a design feature of the Tonal Interval Space to provide a consonance indicator of musical audio, compare to a i) uniformly-weighted space (e.g., $w_u(k) = \{1, 1, 1, 1, 1, 1\}$); ii) the previously proposed weights, $w_s(k) = \{2, 11, 17, 16, 19, 7\}$, adjusted for symbolic music representations [1]; and iii) other metrics adopted in computational mashup creation (namely, psychoacoustic models [17, 20, 9]) in measuring the consonance of common triads. Second, we assess how the perceptual relatedness, R, metric compares to previous harmonic metrics used in computational mashup creation. Finally, we demonstrate the extent to which the harmonic compatibility metric, H, resulting from the combination of the two previous indicators, promotes principles embodied in Western tonal harmony. In all experiments, the pitch content of musical audio is represented using the harmonic template of an "average orchestral instrument" shown in Fig. 2.

In Table 2, we show that the Tonal Interval Space is more consistent in measuring the consonance of common triads than psychoacoustic models used in related mashup literature [20, 17, 9], as it preserves to a higher degree the empirical ranking of triads consonance. Moreover, we demonstrate that the weights, $w_a(k)$, computed in Sec. 2 are decisive in capturing the consonance of musical audio in the Tonal Interval Space, as both a uniformly-weighted space and the previously proposed weights for symbolic music inputs [1] fail at providing a ranking of triads consonance from musical audio in line with empirical ratings.

In Table 3, we show the perceptual relatednesses, R, of dyads in the Tonal Interval Space to which we compare to dyads distances in the (cosine) chroma vector space. The ranking order of dyads perceptual relatedness in the Tonal Interval Space is consistent with tonal harmony principles in the sense it promotes tertian harmony as a result of having fifths and thirds at a closer distance than all remaining intervals. The (cosine) distance between pitch class chroma vectors, adopted as a harmonic compatibility metric in previous computational mashup works [7, 6, 14], largely agrees with the dyads perceptual ranking from the Tonal Interval Space, with the notable exception of the minor seconds and major seventh which are closer in this metric space than the major and minor

Table 2. Ranking order of common triads consonance from empirical data [5], psychoacoustic models [17, 20] and the Tonal Interval Space using uniform weights, $w_u(k)$, the previously symbolic-adjusted weights, $w_s(k)$, and the newly proposed weights for musical audio, $w_a(k)$. 1 corresponds to the most consonant chord and 5 the most dissonant.

	Empirical Data	Psychoacoustic Models		Tonal Interval Space		
Chord quality	Cook et al. [5]	Parncutt [17]	Sethares [20]	$w_u(k)$	$w_s(k)$	$w_a(k)$
major	1	2	2	2	1	1
minor	2	3	2	2	1	1
sus4	3	–	1	1	2	2
dim	4	4	4	3	3	3
aug	5	1	5	2	1	4

Table 3. Dyads distance in the Tonal Interval Space–using the perceptual relatedness, R, metric in Eq. 2, and in a (cosine) space between chroma vectors, $c(n)$. The smaller the distance the more related two dyads are assumed to be. Values are normalized to maximum distance for enhanced readability.

Distance	P1	m2/M7	M2/m7	m3/M6	M3/m6	P4/P5	TT
R	0	1	.94	.84	.88	.77	.94
$c(n)$	0	.91	1	.99	.94	.87	.98

thirds or their complementary minor and major sixth. Consequently, the chroma space disrupts a preference for tertian harmonies.

In Table 4, we show the six best ranking triads resulting from overlapping all dyad combinations (without repetition and order relevance) of the $\{1-11\} \in \mathbb{Z}$ integer set to the C or 0 pitch class (i.e., 55 triads) using four metrics: the cosine distance between chroma vectors, $c(n)$; perceptual relatedness, R; consonance, C; and harmonic compatibility, H. The six best ranking triads in the cosine chroma space and the perceptual relatedness, R, are aligned with the findings shown in Table 3. The chroma space favors chords including P4/P5 and m2/M7 intervals. Conversely, the perceptual relatedness, R, favors chords including P5/P4, m3/M6, and M3/m7 intervals. As such, combining sonorities with small R values, results in extended chords with stacked fifths and thirds. However, while the chords resulting from these vertical aggregates are building blocks of Western tonal harmony, the best ranked chords result in multiple seventh chords (with omitted notes), and not, in the ideally expected, triads (e.g., major, minor, and diminished).

When combining the perceptual relatedness, R, and consonance, C, indicators, we enforce the preference for common major, minor, and suspended fourth triads—the most common building blocks of the Western tonal harmony. Nonetheless, the harmonic compatibility, H, ranking in Table 4 ignores the key level promoting chromaticism (non-diatonic) progressions between neighbor sonorities. To account for this hierarchical level of analysis we provide in our

Table 4. Ranking order of triads resulting from overlapping all dyad combinations to the C or 0 pitch class using four metrics: the cosine distance of chroma vectors, $c(n)$, perceptual relatedness, R, Consonance, C, and harmonic compatibility, H. To the pitch class set of the resulting triads, we include the chord label, whenever unambiguous and complete triads are formed. A musical notation along with sounding examples of the table contents are available at: http://bit.ly/2pGvAnO.

$c(n)$	$0,5,7$ C sus4	$0,1,7$	$0,5,11$	$0,1,5$	$0,7,11$	$0,4,7$ C	$0,5,8$ F-
R	$0,5,7$ C sus4	$0,3,5$	$0,7,9$	$0,7,8$	$0,4,5$	$0,3,7$ C-	$0,5,9$ F
C	$0,3,8$ Ab	$0,3,7$ C-	$0,5,8$ F-	$0,4,7$ C	$0,5,9$ A	$0,4,9$ F	$0,5,7$ F sus4
$\bar{R} \cdot \bar{C}$	$0,5,7$ C sus4	$0,3,7$ Ab	$0,3,8$ C-	$0,5,8$ F-	$0,5,9$ F	$0,4,7$ C	$0,4,9$ A-

interactive visualization a layer of information which can guide users in selecting (in-key) diatonic mixes.

7 Conclusion and Future Work

In this paper we have presented a novel harmonic compatibility metric as a combination of two indicators of perceptual relatedness and consonance computed from the Tonal Interval Space [1]. To this end, we adapted the Tonal Interval Space to provide a robust indicator of tonal consonance for musical audio based on a harmonic spectrum template of an "average orchestral instrument." Our evaluation has shown that the indicators computed from the Tonal Interval Space are aligned with principles of the Western tonal music, as they favor the most common building blocks of tertian harmony with enhanced consonance. An interactive visualization of the harmonic compatibility between musical audio samples from a collection was presented. It includes an extra layer of information which links audio samples to their estimated key as well as exposes the onset density and spectral region as graphical attributes (i.e., color and shape).

In future work, we plan to assess the perceptual basis of the indicators from the Tonal Interval Space using musical audio with different timbral attributes, as well as understand the relevancy of the proposed method for EDM and the practice of the DJ in meaningful world-case application scenarios for assisting their online and offline creative flow.

Acknowledgments. Project TEC4Growth—Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020 is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). This research is also supported by the Portuguese FCT, under the project IF/01566/2015 and post-doctoral grant SFRH/BPD/109457/2015.

References

1. Bernardes, G., Cocharro, D., Caetano, M., Guedes, C., Davies, M.: A multi-level tonal interval space for modelling pitch relatedness and musical consonance. Journal of New Music Research 45(4), 281–294 (2016)
2. Bernardes, G., Cocharro, D., Guedes, C., Davies, M.E.P.: Harmony generation driven by a perceptually motivated tonal interval space. ACM Computers in Entertainment 14(2) (2016)
3. Bernardes, G., Davies, M., Guedes, C.: Audio key estimation with adaptive mode bias. In: Proceedings of ICASSP (2017)
4. Brent, W.: A timbre analysis and classification toolkit for pure data. In: Proceedings of ICMC (2010)
5. Cook, N.: Harmony, Perspective, and Triadic Cognition. Cambridge University Press (2012)
6. Davies, M., Stark, A., Gouyon, F., Goto, M.: Improvasher: A real-time mashup system for live musical input. In: Proceedings of NIME. pp. 541–544 (2014)
7. Davies, M.E.P., Hamel, P., Yoshii, K., Goto, M.: Automashupper: Automatic creation of multi-song music mashups. IEEE Trans. ASLP 22(12), 1726–1737 (2014)
8. Euler, L.: Tentamen novae theoriae musicae. Broude (1968/1739)
9. Gebhardt, R., Davies, M., Seeber, B.: Psychoacoustic approaches for harmonic music mixing. Applied Sciences 6(5) (2016)
10. Griffin, G., Kim, Y., Turnbull, D.: Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. In: Proceedings of ICASSP. pp. 2–5 (2010)
11. Huron, D.: Interval-class content in equally tempered pitch-class sets: Common scales exhibit optimum tonal consonance. Music Perception 11(3), 289–305 (1994)
12. Johnson-Laird, P.N., Kang, O.E., Leong, Y.C.: On musical dissonance. Music Perception 30(1), 19–35 (2012)
13. Jordà, S., Kaltenbrunner, M., Geiger, G., Alonso, M.: The reactable: a tangible tabletop musical instrument and collaborative workbench. In: ACM SIGGRAPH 2006 Sketches. p. 91. ACM (2006)
14. Lee, C.L., Lin, Y.T., Yao, Z.R., Lee, F.Y., Wu, J.L.: Automatic mashup creation by considering both vertical and horizontal mashabilities. In: Proceedings of ISMIR. pp. 399–405 (2015)
15. Manovich, L.: The Language of New Media. MIT press (2001)
16. Mixed in Key: Mashup 2, `http://mashup.mixedinkey.com`, last accessed on 28 March 2017
17. Parncutt, R.: Harmony: A Psychoacoustical Approach. Springer (1989)
18. Plazak, J., Huron, D., Williams, B.: Fixed average spectra of orchestral instrument tones. Empirical Musicology Review 5(1) (2010)
19. Puckette, M.: Pure data. In: Proceedings of ICMC. pp. 224–227 (1996)
20. Sethares, W.: Tuning, Timbre, Spectrum, Scale. Springer-Verlag (1999)
21. Sha'ath, I.: Estimation of key in digital music recordings. Master's thesis, Birkbeck College, University of London (2011)
22. Shiga, J.: Copy-and-persist: The logic of mash-up culture. Critical Studies in Media Communication 24(2), 93–114 (2007)
23. Zwicker, E., Fastl, H.: Psychoacoustics–Facts and Models. Springer Verlag (1990)

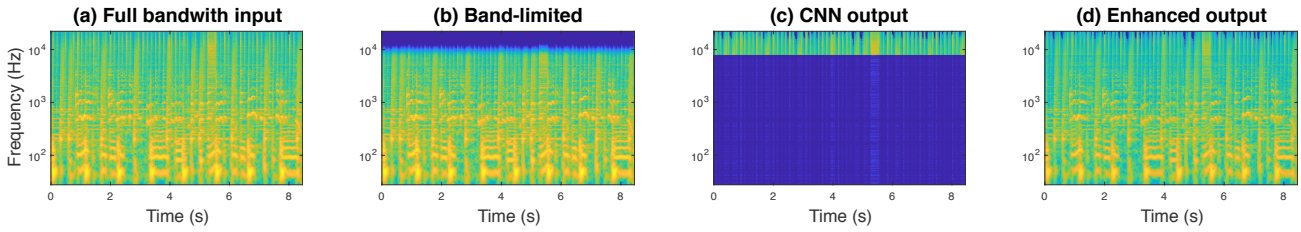


Figure 1: *Illustrative overview of our proposed approach for bandwidth extension. (a) The CQT of a short musical audio input sampled at* 44.1 *kHz. (b) The band-limited version resulting from a low-pass filter with a cutoff frequency of* 7500 *Hz. (c) The high frequency output of the CNN*[1]*. (d) The enhanced output signal obtained by combining the band-limited and CNN reconstruction.*

as the axes of an image, we explore two musically motivated CNN architectures: bottleneck and stride [13, 14] rather than more standard square filters in image processing.

For our musical inpainting problem, we aim to reconstruct or "complete" a strip covering the highest frequency bins of a time-frequency, for which an illustrative example is shown in Figure 1. While this is conceptually related to the idea of filling temporal gaps (*i.e.*, missing vertical strips) [15, 16] these methods exploit temporal redundancy via repetition in the musical input, where as in our approach, the high frequency region is never observed.

A particular novelty of our proposed approach is to leverage implicit knowledge of musical structure by the use of the constant-Q spectrogram. For bandwidth extension, the CQT has a potentially advantageous property over the STFT, which is that, due to the logarithmic spacing of the CQT bins, we can make a richer observation of the narrow-band (*i.e.*, low-frequency) region in order to reconstruct a smaller amount of higher frequency information. Comparing the STFT and CQT in matrix form (where rows correspond to frequency and the columns to time) this means that for an identical cut-off frequency (*e.g.*, of $f_s/4$), and a roughly equal total number of frequency channels, a far smaller amount of data must be reconstructed for the CQT than for the STFT. Until recently, such potential benefits remained theoretical due to the absence of an inverse CQT transform. However, recent work leveraging the non-stationary Gabor transform (NSGT) [17, 18] has demonstrated that perfect reconstruction of the CQT is both possible and executable in reasonable computation time.

For this initial work, our primary focus is towards the reconstruction of magnitude spectrograms, thus we do not attempt any automatic reconstruction of the phase spectrogram. Instead we make use of the original phase from the band-limited version, without any subsequent modification. Our evaluation focuses on the measurement of the signal to distortion ratio (SDR) for the enhanced and band-limited versions. In this way, the extent of the enhancement provided by our approach can be assessed by the increase in SDR over the band-limited versions.

The remainder of this paper is structured as follows. In Section 2 we contrast our approach with existing work in audio bandwidth extension. In Section 3, we detail our proposed method using convolutional neural networks, which we evaluate in Section 4, and provide discussion and conclusions in Section 5.

[1]While the CNN outputs a full wide-band spectrogram, the region below the cut-off has been attenuated for greater visual clarity.

2. RELATION WITH PREVIOUS WORK

With the exception of [5, 9, 19], most previous research in audio bandwidth extension has been applied to speech signals. Regarding the methodology, the deep learning approaches in [8, 9] are the closest to our proposed method. In the same way as [9], which uses a similar approach to image super-resolution [20], we are inspired by recent advancements in image processing using CNNs [10, 11]. Unlike [7] we eliminate all accompanying heuristics and estimate the high-frequency spectra directly with the neural networks.

In contrast to the NMF speaker-specific spectral bases used in [3, 19] or the codebook of the LPC approach [2], we are concerned with the generalization capabilities of our trained model and do not seek to tailor our approach for specific individual pieces of music. Furthermore, we do not tune any method-specific hyper-parameters or weighting coefficients which were previously used in [2] as a part of a chain of signal processing heuristics.

Similar to the convolutional NMF approach in [3], the hidden Markov models (HMM) in [21], and the time-series CNN in [9], we consider cross-frame contextual dependencies. These short-term dependencies are learned by CNNs using horizontal filters for a given time-context, while timbre features are learned using vertical filters [13, 14].

The CNN approaches used in image restoration, completion, or inpainting [22, 10, 11] are exposed to the entire image and not just to the missing patches in order to perform the reconstruction. In a similar fashion, we use the observation of the lower frequencies to better reconstruct the higher frequencies.

3. METHOD

3.1. Overview

An overview of our proposed method, which comprises two stages: training and enhancement, can be seen in Figure 2. For training we require a dataset comprising full-bandwidth music recordings and narrow-band versions which lack high frequency content above a specific cutoff frequency. We obtain narrow-band versions by applying a low-pass filter to the original recordings. Then, we compute the desired time-frequency representation, using the STFT or CQT, and extract the respective magnitude spectrogram for each channel of the stereo recordings. Additionally, we apply the data processing heuristics described in [23] and train the CNNs with the architectures described in Section 3.3 and the training procedure in Section 3.4.

The enhancement stage is detailed in the Section 3.5, where the high-frequency content is obtained by feeding the magnitude

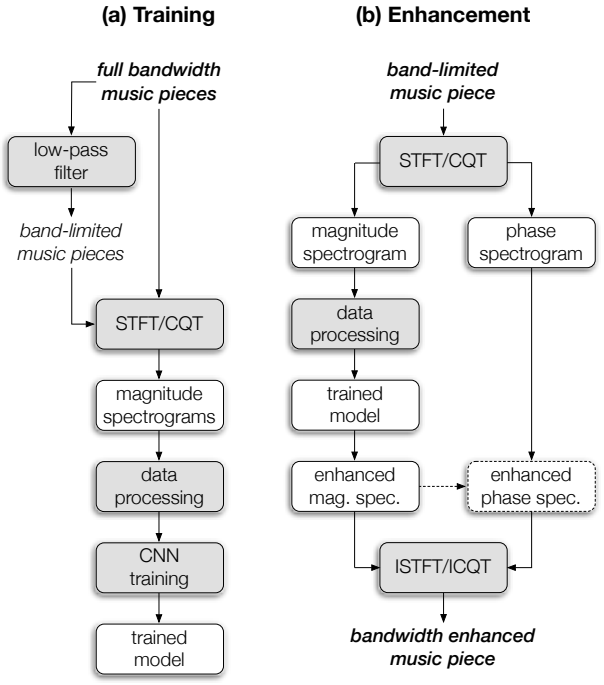


Figure 2: *Overview of our bandwidth extension system. (a) The training stage has access to full-band and band-limited music signals. (b) The enhancement stage only observes the band-limited signals. Boxes shaded in grey indicated processes, where as those in white correspond to data. The term data processing is used to encapsulate the partitioning of the data into overlapping chunks. The dashed arrow and box indicate optional processing which is not undertaken in this work.*

spectrograms forward through the previously trained CNN. The phase spectrogram of the band-limited version is retained to compute the inverse STFT or CQT.

3.2. Feature computation

We calculate the STFT or the CQT [18] of the stereo audio mixture as $\mathbf{X}_i(t, f)$ where $i = 1, 2$ are the stereo channels, t is the time axis and f is the frequency axis. In order to focus on the reconstruction of the magnitude spectrum, we discard the phase when computing the training features for the neural network.

The CNN architectures used in this paper require a fixed input size (T, F), where T is the temporal context in time frames and F is the total number of frequency bins corresponding to the STFT or CQT magnitude spectrograms. To obtain magnitude spectrograms of fixed duration, the variable-size magnitude spectrograms of each music piece are split into overlapping chunks of fixed size T time frames with an overlap of O frames. In addition, splitting the input signal into chunks leads to a smaller network, with fewer parameters to train, and thus a lower computational burden. These data processing heuristics adopted prior to training are described in detail in [23] and were used previously for the task of audio source separation for full length musical recordings [14, 23, 24].

3.3. Convolutional autoencoders

We present two musically motivated CNN autoencoder architectures, the CNN bottleneck in Section 3.3.1 and the CNN stride-2 in Section 3.3.2. Since time and frequency in magnitude spectrograms have different meanings than the horizontal and vertical axes in images, we should not adopt image-processing square filters. Instead, we follow [13, 14] by using vertical filters to model frequency components and horizontal filters to model their temporal evolution. A further distinction is that the magnitude spectrograms of audio signals are sparse [25]. Thus, we use a sparse activation function between the layers, specifically, rectified linear units (ReLU) [26]. In addition, the CNN bottleneck architecture has a dense bottleneck layer with a low number of units to compress, or reduce, the learned features. On a related note, the CNN stride-2 architecture comprises successive convolutions with a stride[2] of two which is the equivalent of learning features by successively downsampling the inputs by a factor of two.

The inputs to both the CNN architectures are multiple magnitude spectrograms of size (T, F), across the channel dimension i. In our case, the learned feature maps are shared between the two input channels [26]. We argue that the CNN can learn more diverse filters from music mixtures with a wide stereo image and therefore we provide magnitude spectrograms for both channels as input. In a further parallel with image processing, this can be considered similar to using the RGB layers of colour images rather a single greyscale image.

The CNN autoencoders comprise an encoding and a decoding stage. The encoding stage contains convolutional and feed-forward layers, while the decoding stage performs the inverse operations of the convolutions in the reverse order such that the output of the CNN has the same dimensions as its input, $(2, T, F)$. Note, we do not use a soft-mask as in music source separation, but instead we directly estimate the magnitude spectrogram with enhanced high-frequency content, $\hat{\mathbf{X}}$. In addition, we assume that the frequency content to be recovered does not have higher energy than the low frequency content. To this end, we limit all the values of $\hat{\mathbf{X}}_i(t, f)$ to the maximum value in channel i at time frame t of the input $\mathbf{X}_i(t, f)$.

3.3.1. CNN bottleneck

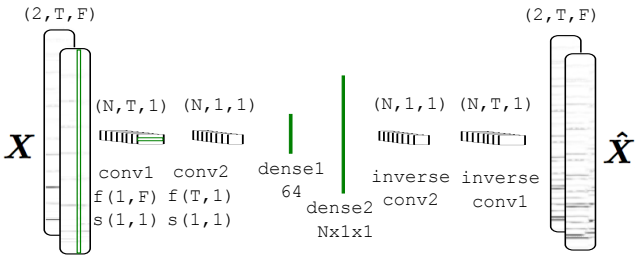


Figure 3: *CNN bottleneck autoencoder architecture [14]. For each layer we give the shape of the filters, strides and feature maps.*

We test a version of the CNN bottleneck successfully used in music source separation [14, 24, 23]. A diagram of the architecture is depicted in Figure 3, and comprises a horizontal convolution, *conv1*, a vertical convolution *conv2*, a bottleneck dense layer

[2]The stride controls how much a filter is shifted on the input.

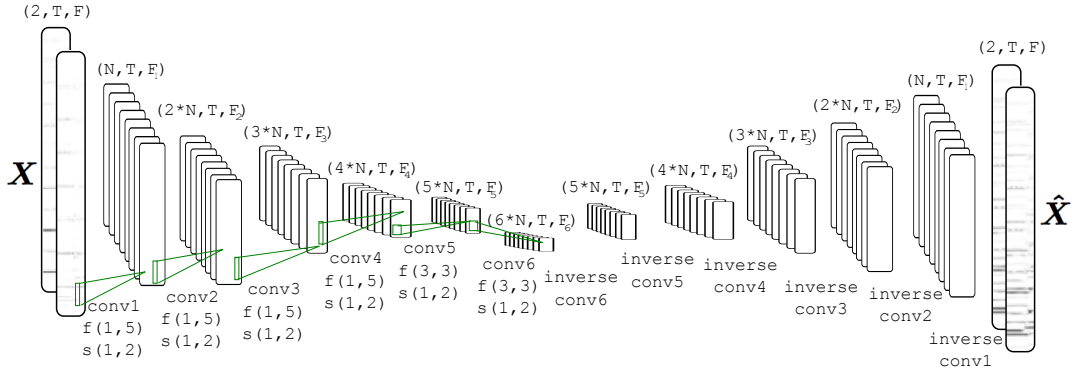


Figure 4: *CNN stride-2 autoencoder architecture. For each layer we give the shape of the filters, strides and feature maps.*

dense1, and another dense layer *dense2* to recover the dimensionality needed to perform the inverse operations of *conv2* and *conv1*. We have N filters for $conv1$ and $conv2$.

3.3.2. CNN stride-2

Small successive convolutional layers with a stride of two have been shown to reduce the number of parameters in a network [27]. Therefore, in contrast to the CNN bottleneck, we target a deep architecture comprising small convolutions. Moreover, time-frequency representations of musical signals often exhibit evenly spaced harmonic components. By modeling frequency content in strides of two we aim to capture high frequency harmonics learned from their low frequency counterparts.

An overview of the stride-2 architecture is shown in Figure 4. For each layer k, the feature maps reduce their frequency size: $F_k = (F_{k-1} - 5)/2 + 1$, as explained in [24]. We have four successive $(1, 5)$ convolutions in frequency, followed by two, two-dimensional $(3, 3)$ convolutions to capture the time-frequency dependencies, each considering the reduction performed by the previous layers.

3.4. Training procedure

Although the output of the CNN, $\hat{\mathbf{X}}$, contains a reconstruction of the magnitude spectrogram across all frequency bins, the parameters of the autoencoder are trained according to a loss function which only considers the reconstruction in higher frequencies. Thus, the loss function L_c depends on the cutoff frequency in bins c and is defined in equation (1) as the mean-squared error (MSE) between the target magnitude spectrogram $\bar{\mathbf{X}}$, and the estimated magnitude spectrograms, $\hat{\mathbf{X}}$:

$$L_c = \sum_{t,f,i} \|u(f-c)(\bar{\mathbf{X}}_i(t,f) - \hat{\mathbf{X}}_i(t,f))\|^2, \qquad (1)$$

where $u(f - c)$ is the unit step function which is 0 for the bins lower than c and 1 for the bins greater than or equal to c.

The parameters of the CNN are updated according to the loss function L_c using mini-batch Stochastic Gradient Descent with the *Adamax* algorithm [28].

3.5. Enhancement

When computing the STFT or CQT for enhancement, we retain the phase and we split the magnitude spectrogram into overlapping chunks of size T time frames with an overlap of O frames as in the training stage. For each chunk $\mathbf{X}$ we obtain an estimation $\hat{\mathbf{X}}$. We then use the estimated chunks to reconstruct the enhanced magnitude spectrogram through the overlap-add procedure as described in [23] and as used in [14, 23, 24].

In contrast to deep learning source separation methods, the estimated spectrogram is not the result of Wiener filtering [29] which ensures that the spectrograms of the sources sum to the input spectrogram. Instead, we need to ensure that the original low-bandwidth content is preserved. To this end, we blend the high-frequency part of the estimations yielded by the network, $\hat{\mathbf{X}}$, with the low-frequency part of the input, $\mathbf{X}$:

$$\tilde{\mathbf{X}}_i(t,f) = (1 - r_c(f))\mathbf{X}_i(t,f) + r_c(f)\hat{\mathbf{X}}_i(t,f) \qquad (2)$$

where $r_c(f) = \max(0, \min(1, f - c))$ is a ramp function depending on the the cutoff frequency in bins c.

As specified in Section 1, we only attempt to reconstruct the magnitude spectrum – without access to phase information when training. However, in order to invert either the reconstructed STFT or CQT we must provide phase information. To this end, we use the phase spectrogram from the band-limited version, as shown in Figure 1(b). Finally, the bandwidth extended audio signals are obtained using with an inverse overlap-add STFT or inverse CQT [18].

4. EVALUATION

The basis of our evaluation is to compare the reconstruction from the STFT and CQT, with the two different CNN autoencoder models: bottleneck and stride-2, and across two cutoff frequencies of 3500 Hz and 7500 Hz. In total, this creates eight reconstruction conditions for comparison.

4.1. Experimental setup

We test our approach on the publicly available Medleydb dataset [30] comprising 121 multi-tracks from which we use the stereo mixes (in uncompressed .wav format sampled at 44.1 kHz and with 16-bit resolution). The dataset covers the following genres: Singer/Songwriter, Classical, Rock, World/Folk, Fusion, Jazz,

Pop, Musical Theatre, Rap. There are 52 instrumental tracks and 70 tracks containing vocals. We randomly split the dataset in training and testing subsets with a ratio of 0.8 (*i.e.*, 80% for training and 20% for testing).

4.1.1. Evaluation metrics

As the basis for the evaluation, we use the BSS_Eval framework [31], a widely used tool to objectively evaluate the quality audio source separation. Within BSS_Eval, the *Source to Distortion Ratio* (SDR) measures the distortion between a target and the estimated multi-channel audio sources. With respect to high-frequency reconstruction, BSS_Eval gives more weight to lower frequency bands and penalizes more frequency content which is not in the target audio, even though this content might be perceptually relevant. In this sense, we recognise that a subjective listening experiment would be a critical important component of future work, but for this initial research, we adopt the SDR as our primary objective measure for this context. It is important to note that we exclude other metrics related to the artifacts, interference, and spatial distortion from BSS_Eval as these are designed particularly for source separation. The SDR is reported for each of the overlapping chunks of 30 seconds with a 15 second overlap.

4.1.2. Time-frequency transform parameterisation

The STFT is computed using a Hann window of length 1024 samples, which at a sampling rate of 44.1 kHz corresponds to 23.2 milliseconds (ms), and a hop size of 512 samples (11.6 ms).

The CQT is computed with the MATLAB toolbox in [18] using the default parameterization, with a minimum frequency of 27.5 Hz, and a frequency resolution of 48 bins per octave. Up to the Nyquist rate of 22.05 kHz this gives 463 logarithmically-spaced frequency bins. Perfect reconstruction via the inverse CQT comes at the expense of high redundancy in time and results in 647 time frames per second, *i.e.*, a temporal resolution of 1.5 ms which is much finer than that of the STFT, while retaining a similar number of frequency bins (463 compared to 513).

Since our goal is to reconstruct the higher frequency end of the magnitude spectrograms, we must contend with the fact that signal energy typically is much lower at higher frequencies than at the lower end. In the context of our convolutional neural network approach this creates a difficulty, since the high frequency magnitude spectrum we seek to predict may have very small values. To partially circumvent this issue, we can apply a logarithmic scaling to both the STFT and CQT magnitude spectrograms prior to training (and subsequently revert back to linear magnitude scaling prior to the eventual output signal reconstruction). However, before applying such a logarithmic scaling we must ensure all magnitude spectrum values (for both the STFT and CQT) are greater than 1, since any values below 1 will be negative after taking the logarithm, and thus ignored by the ReLU. To this end we apply the logarithmic scaling as follows: $\mathbf{X}_{\log} = \log_{10}(\alpha + \beta\mathbf{X})$, where $\mathbf{X}$ refers to either the STFT or CQT. For the CQT we set $\alpha = 1$ and $\beta = 4$, where as for the STFT no scaling is required thus we set $\alpha = 1$ and $\beta = 1$. The final stage of the pre-processing relates to deep learning methods usually requiring data to be normalized to an interval or include a batch-normalization step. Thus, we normalize all the training data to be between 0 and 1 by multiplying with a scale factor, which we set as the maximum of the training data.

To create the band-limited, *i.e.*, low-pass filtered versions of the music pieces for training (and subsequent reconstruction), we use an 8^{th} order Butterworth filter. In order to explore two different conditions, we create one low-pass filtered version with a cutoff of 3500 Hz and another at 7500 Hz (approximately $f_s/12$ and $f_s/6$). For both, we seek to reconstruct the full remaining frequency range of the original recordings up to the Nyquist rate of 22.05 kHz).

We split the STFT or CQT into overlapping chunks of $T = 30$ time frames with an overlap of $O = 10$. Chunks are randomly grouped each epoch into batches of 32. For a fair comparison between bottleneck and stride-2 we use $N = 175$ of filters for bottleneck and $N = 40$ filters for stride-2, such that the number of parameters is equal for both of the architectures (1.8 million). The STFT is trained for 100 epochs. Since CQT has a higher time resolution, we generate more training data and we only train the network for 32 epochs. The initial learning rate is 0.001 for STFT and 0.0001 for CQT.

4.1.3. Implementation details

The code used in this paper is built on top of Pytorch, a framework for neural networks [3]. We ran the experiments on an Ubuntu 16.04 PC with GeForce GTX TITAN X GPU, Intel Core i7-5820K 3.3GHz 6-Core Processor, X99 gaming 5 x99 ATX DDR44 motherboard. Training a condition took 16 hours for the STFT and 44 hours for the CQT; by contrast, the enhancement stage runs faster than real-time on the same hardware. To ensure reproducibility, a fixed seed controls the pseudo-random number generation in Python. This is used when initialize the parameters of the CNN and to randomly split the dataset into training and testing. The results presented in Section 4.2 are for seed 0.

4.2. Results

The results for the bottleneck and stride-2 are shown in terms of SDR in Figure 5a and 5b for the CQT and STFT respectively. In each figure we present the SDR across the cutoff frequencies of 3500 Hz and 7500 Hz and show the difference in performance for examples in the training set versus those withheld for testing. Since we want to measure how much the quality of the reconstruction improves with respect to the low-pass input, we include the SDR for all the low-pass versions of the pieces in the dataset.

On inspection of the figures we can see that the best overall performance for the test set is obtained using the stride-2 architecture for the cutoff of 3500 Hz and the bottleneck architecture for the cutoff of 7500 Hz. In both of these conditions there is a negligible difference between the SDR on those musical recordings used for training, compared to those withheld for testing. In addition to the highest overall mean SDR values, we can additionally observe the greatest relative difference over the mean SDR of the low-pass filtered versions. For both approaches there is a relative increase in SDR of over 4 dB. Since the SDR calculation is made directly on the waveforms, this suggests that relevant high frequency information from the original recordings is being reconstructed based soley on observing the band-limited versions.

When looking across the two architectures for the CQT results, we can observe that the stride-2 approach is less effective for the higher cutoff of 7500 Hz. This may be due to the lower proportion of harmonic content above this cutoff, and hence the reduced impact of the stride's ability to model harmonic relationships.

[3] `http://pytorch.org`

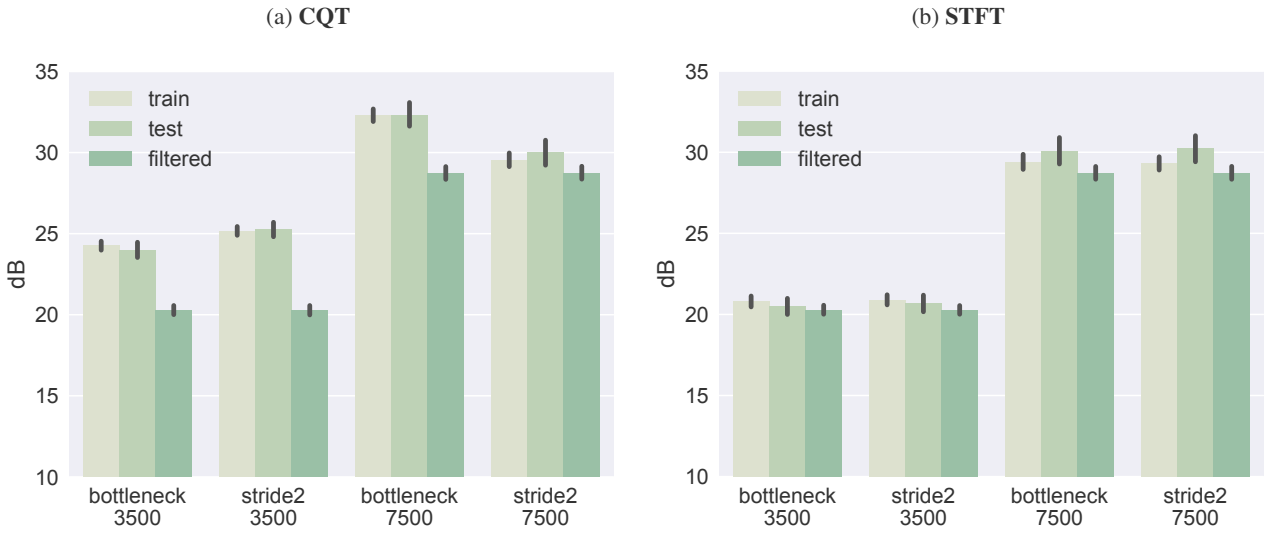


Figure 5: *SDR for (a) CQT and (b) STFT representations. The results compare the difference in SDR for training and testing sets, and the low-pass filtered condition (without enhancement), for the bottleneck and stride-2 CNN architectures and the cutoff frequencies of* $3500\,Hz$ *and* $7500\,Hz$*. The black vertical lines represent the 95% confidence intervals.*

Looking at the comparison between the CQT and STFT, we can identify two main differences. First, the absolute SDR for the STFT enhanced versions are lower than for the CQT across all conditions, and in turn, the relative improvement over the low-pass filtered versions is also reduced. This behaviour is in line with our original hypothesis concerning the advantage of using the CQT, where, although the frequency range to reconstruct is the same for both time frequency representations, the number of missing rows of the CQT is far smaller than that of the STFT. This is also consistent with results from image completion, in which larger image patches are more difficult to recover than smaller ones [10]. Another important factor may be the difference in temporal resolution for the two time-frequency representations, which is greater by a factor of approximately 8 to 1 for the CQT compared to the STFT; that while both process overlapping chunks of $T = 30$ time frames, the reconstruction of the CQT is much more localised in time than the STFT. We intend to explore this effect in future work by increasing the frame overlap in the STFT to a comparable level to that of the CQT. However, any significant increase in the frequency resolution of the STFT, *e.g.*, by using a larger window size would drastically increase the size of the model to be trained, and thus negate the approximately equal number of frequency channels in the STFT and CQT in our current setup.

To complement these objective results, we provide a set of short sound examples covering the eight reconstruction conditions, together with the original and two low-pass filtered versions. Furthermore, for the two best performing conditions: CQT stride-2 $3500\,$Hz and CQT bottleneck $7500\,$Hz we provide an informal comparison of different approaches for phase reconstruction. To this end, we include phase reconstruction using: i) the low-pass filtered version (our proposed method); and ii) using low-pass filtered version below the cutoff and random phase above it. All of the sound examples are available at the following website: `http://telecom.inesctec.pt/~mdavies/dafx18/`

5. DISCUSSION AND CONCLUSIONS

We presented a new deep learning method to reconstruct the high frequency content of music recordings. Our evaluation demonstrates that due to to the logarithmic spacing of frequencies, the CQT offers a better time-frequency representation for this problem than STFT in terms of SDR. It is important to stress that these are initial experiments are performed under highly controlled conditions. Due to the high computational cost of training (which took several days using powerful GPUs), we only explored two cutoff frequencies, and used the same type of low-pass filter throughout. On this basis, we do not have sufficient evidence about the generalisation capacities of our trained networks to function under more arbitrary filtering conditions. This is especially important when considering our long term goal of the restoration of old recordings, for which we cannot assume any specific filtering conditions. Furthermore, in this scenario no stereo version of the recording may exist, which would require additional modifications to our approach.

Another important constraint within this study was the treatment of the phase in the reconstruction. While we do not provide unobservable information (*e.g.*, the phase of the original, full-band signal), our approach for using the low-pass filtered version phase could almost certainly be improved via the use of phase reconstruction techniques [32]. Since these are typically applied for an STFT-like representation, we intend to explore the means for doing this directly for in the invertible CQT representation in future work. Furthermore, we recognise the potential of using other time-frequency representations – provided that there is a method to invert them, *e.g.*, using Wavenet as a vocoder [33]. Furthermore, generative adversarial networks have recently became popular in image recovery and super-resolution [10] and can synthesize more realistic time-frequency content, which may yield further improvements to the quality of the signal reconstruction.

With respect to the evaluation, we acknowledge that BSS_Eval

has been primarily designed for audio source separation, and further perceptual experiments are needed to better understand the subjective performance of our proposed method. Furthermore, BSS_Eval metrics do not always correlate with the perceived quality of separation [34]. In contrast to magnitude spectrograms, reconstructed images can be evaluated more directly because the inherent structure in the pixels can be understood in terms of the geometric and textural properties of scenes and objects. However in our approach the images correspond to time-frequency representations which are non-trivial for non-experts to visually interpret, and require an additional transformation stage to be audible. Within our training stage, the loss function relates to the mean squared error between the original magnitude spectrogram and the reconstruction, however our objective evaluation measures the SDR of the reconstructed audio signals, which explicitly includes phase information. Thus, we also intend to explore alternative loss functions (perhaps by using phase information directly) and subsequently investigate their correlation with perceptual ratings of audio quality from trained listeners. As part of this comparison we we intend to incorporate existing approaches for bandwidth extension which have been shown to be effective for music signals sampled at 44.1 kHz.

6. ACKNOWLEDGMENTS

We wish to thank the anonymous reviewers for their expertise and generous feedback which improved the quality of this paper.

Matthew E.P. Davies is supported by Portuguese National Funds through the FCT – Foundation for Science and Technology, I.P., under the project IF/01566/2015. The TITANX used for this research was donated by the NVIDIA Corporation. "TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-00020" is financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF).

7. REFERENCES

[1] H. Yasukawa, "Signal restoration of broad band speech using nonlinear processing," in *European Signal Processing Conference*, 1996, pp. 1–4.

[2] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, "Speech enhancement via frequency bandwidth extension using line spectral frequencies," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2001, vol. 1, pp. 665–668.

[3] D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrowband speech using non-negative matrix factorization," in *Ninth European Conference on Speech Communication and Technology*, 2005, pp. 1505–1508.

[4] E. Larsen and R. M. Aarts, *Audio bandwidth extension: application of psychoacoustics, signal processing and loudspeaker design*, John Wiley & Sons, 2005.

[5] D. L. Sun and R. Mazumder, "Non-negative matrix completion for bandwidth extension: A convex optimization approach," in *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2013, pp. 1–6.

[6] Martin Dietz, Lars Liljeryd, Kristofer Kjorling, and Oliver Kunz, "Spectral band replication, a novel approach in audio coding," in *Audio Engineering Society Convention 112*. Audio Engineering Society, 2002.

[7] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 3, pp. 873–881, 2007.

[8] K. Li, Z. Huang, Y. Xu, and C-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015, pp. 2578–2582.

[9] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," *arXiv preprint arXiv:1708.00853*, 2017.

[10] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 107:1–107:14, 2017.

[11] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6721–6729.

[12] J. C. Brown, "Calculation of a constant q spectral transform," *The Journal of the Acoustical Society of America*, vol. 89, no. 1, pp. 425–434, 1991.

[13] J. Pons, T. Lidy, and X. Serra, "Experimenting with musically motivated convolutional neural networks," in *14th International Workshop on Content-Based Multimedia Indexing (CBMI)*, 2016, pp. 1–6.

[14] P. Chandna, M. Miron, J. Janer, and E. Gómez, "Monoaural audio source separation using deep convolutional neural networks," in *International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*, 2017, pp. 258–266.

[15] T. Jehan, *Creating music by listening*, Ph.D. thesis, Massachusetts Institute of Technology, School of Architecture and Planning, Program in Media Arts and Sciences, 2005.

[16] N. Perraudin, N. Holighaus, P. Majdak, and P. Balazs, "Inpainting of long audio segments with similarity graphs," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018, *(In Press)*.

[17] G. A. Velasco, N. Holighaus, M. Dörfler, and T. Grill, "Constructing an invertible constant-Q transform with nonstationary Gabor frames," *14th International Conference on Digital Audio Effects (DAFx-11)*, pp. 93–99, 2011.

[18] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler, "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution," in *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

[19] P. Smaragdis and B. Raj, "Example-driven bandwidth expansion," in *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2007, pp. 135–138.

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European Conference on Computer Vision*, 2014, pp. 184–199.

[21] P. Jax and P. Vary, "Artificial bandwidth extension of speech signals using MMSE estimation based on a hidden Markov model," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003, vol. 1, pp. I–680–I–683.

[22] X. Mao, C. Shen, and Y-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in neural information processing systems*, 2016, pp. 2802–2810.

[23] M. Miron, J. Janer, and E. Gómez, "Generating data to train convolutional neural networks for classical music source separation," in *14th Sound and Music Computing Conference*, 2017, pp. 227–233.

[24] M. Miron, J. Janer, and E. Gómez, "Monaural score-informed source separation for classical music using convolutional neural networks," in *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017, pp. 55–62.

[25] M. D. Plumbley, T. Blumensath, L. Daudet, R. Gribonval, and M. E. Davies, "Sparse representations in audio and music: from coding to source separation," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

[26] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[27] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

[30] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello, "Medleydb: A multitrack dataset for annotation-intensive MIR research," in *15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014, pp. 155–160.

[31] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[32] Z. Průša, P. Balazs, and P. L. Søndergaard, "A noniterative method for reconstruction of phase from STFT magnitude," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1154–1164, 2017.

[33] M. Blaauw and J. Bonada, "A neural parametric singing synthesizer," *arXiv preprint arXiv:1704.03809*, 2017.

[34] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann, "Subjective and objective quality assessment of audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2046–205, 2011.

[35] Christian R Helmrich, Andreas Niedermeier, Sascha Disch, and Florin Ghido, "Spectral envelope reconstruction via igf for audio transform coding," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 389–393.

BREAK-INFORMED AUDIO DECOMPOSITION FOR INTERACTIVE REDRUMMING

Patricio López-Serrano[1] **Matthew E. P. Davies**[2] **Jason Hockman**[3]
Christian Dittmar[1] **Meinard Müller**[1]
[1] International Audio Laboratories Erlangen, Germany
[2] INESC TEC, Sound and Music Computing Group, Portugal
[3] DMT Lab, Birmingham City University, UK
patricio.lopez.serrano@audiolabs-erlangen.de

ABSTRACT

Redrumming or *drum replacement* is used to substitute or enhance the drum hits in a song with one-shot drum sounds obtained from an external collection or database. In an ideal setting, this is done on multitrack audio, where one or more tracks are dedicated exclusively to drums and percussion. However, most non-professional producers and DJs only have access to mono or stereo downmixes of the music they work with. Motivated by this scenario, as well as previous work on decomposition techniques for audio signals, we propose a step towards enabling full-fledged redrumming with mono downmixes.

1. PROPOSED METHOD

Figure 1 gives an overview of our method for break-informed redrumming, inspired by [5]. The figure consists of two sides: the left side contains the track that will be redrummed, and the right side contains the track that will provide new timbral information. We describe each side in the following.

As input for the track that will be redrummed, we need the monaural (mono) downmix of a song that contains a drum break, as well as a segmentation indicating the location of the drum break. A method for automatically finding percussion-only regions in digital music recordings is described in [4]. Since we are working with mono input data, we need a way to extract multiple drum kit tracks from the mixture. At this stage we use the drum source separation (DSS) method by Dittmar and Müller [1] (which is based on non-negative matrix factor deconvolution, NMFD) to learn timbral templates for the drum kit elements present in the break.

Following the assumption that the drum timbre remains largely unchanged throughout the track, we fix the drum templates and learn a further non-negative matrix factorization (NMF) for the entire track, following [2]. In principle, this is equivalent to NMF-based harmonic-percussive source separation (HPSS), such as recently proposed by [3]. We now have spectral information for all the non-percussive instruments (or the *harmonic* part, shown as horizontal orange bars), which includes lead melody as well as accompaniment instrumentation. Our NMF model has also learned activations for the fixed drum templates—these activations are shown as light purple curves for kick drum (KD), snare drum (SD), and hi-hat (HH), and will be used as insertion positions for the redrumming.

The right side of Figure 1 shows a second song containing a drum break, along with an indication of where the break is. This drum break will provide the timbral information for the redrum, with sounds being "copied" onto appropriate timepoints in the left-hand track. Again, we use NMFD-based DSS [1] to learn the timbral properties of individual drum hits in the break, in order to combine these spectral templates with the activations found in the left-hand track. We encourage listening to audio examples of our results at the accompanying website. [1]

2. REAL-WORLD SCENARIO

Fully automatic redrumming is a difficult task. Especially when working with audio mixtures, the main challenge lies in avoiding crosstalk—not only between the harmonic and percussive parts, but also among the individual drum kit parts themselves. In a studio setting, redrumming is usually done by selecting (one-shot) drum hit sounds from a collection or database—in this contribution, instead of having a ready-made drum sound database, we construct this collection from a drum break of the user's choosing. Thus, for instance, we can imagine that a user selects James Brown's "Funky Drummer" as a track to be redrummed, using the sounds (or timbral properties) from "Amen, Brother" by The Winstons.

Redrumming is also usually done manually or semi-manually, using a DAW such as `Logic Pro` or `Cubase`, or a specialized plugin, like `Drumagog` or `Superior Drummer`. Assuming that a user wishes to substitute KD

[1] `https://www.audiolabs-erlangen.de/resources/MIR/2018-ISMIR-LBD-Redrum`

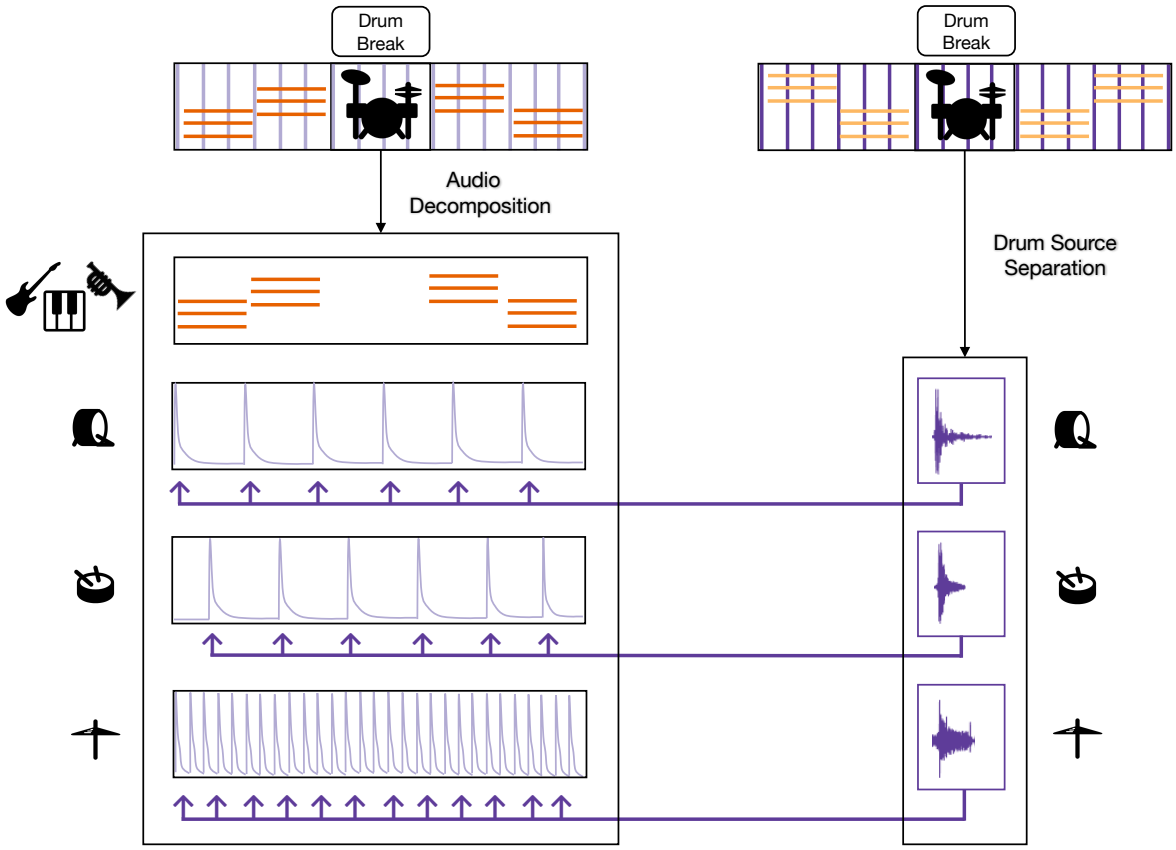


Figure 1. Overview of our break-informed redrumming method. For a detailed explanation, see Section 1.

hits, the procedure would be to step through all the transients in the drum track, placing the cursor at the beginning of each KD hit. Then, the user would select a sound from the library and use it to either replace the current KD hit, or place it in a new track to enhance the timbral properties of the existing one. An advantage of our system is that all hits of a certain type (e.g., all KD hits) are tracked simultaneously by our decomposition model, enabling a once-through automatic substitution.

Furthermore, in a professional setting, musicians often request a certain *sound* or *aesthetic* for the mixing process, wishing to sound like other artists that they admire. Thus, another advantage of our method is that users can directly achieve this effect by selecting a drum break recording with their desired properties.

3. ACKNOWLEDGMENTS

Project TEC4Growth - Pervasive Intelligence, Enhancers and Proofs of Concept with Industrial Impact/NORTE-01-0145-FEDER-000020, financed by the North Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, and through the European Regional Development Fund (ERDF). MD is supported by national funds through the FCT - Foundation for Science and Technology, I.P., under the project IF/01566/2015. PLS is supported in part by a scholarship from CONACYT-DAAD. CD and MM are supported by the German Research Foundation (DFG-MU 2686/10-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institute for Integrated Circuits IIS.

4. REFERENCES

[1] Christian Dittmar and Meinard Müller. Reverse engineering the amen break – score-informed separation and restoration applied to drum recordings. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(9):1531–1543, 2016.

[2] Minje Kim, Jiho Yoo, Kyeongok Kang, and Seungjin Choi. Nonnegative matrix partial co-factorization for spectral and temporal drum source separation. *IEEE Journal of Selected Topics Signal Processing*, 5(6):1192–1204, 2011.

[3] Clément Laroche, Matthieu Kowalski, Hélène Papadopoulos, and Gaël Richard. Hybrid projective nonnegative matrix factorization with drum dictionaries for harmonic/percussive source separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1499–1511, 2018.

[4] Patricio López-Serrano, Christian Dittmar, and Meinard Müller. Finding drum breaks in digital music recordings. In *Proceedings of the International Symposium on Computer Music Multidisciplinary Research (CMMR)*, pages 68–79, Porto, Portugal, September 2017.

[5] Tomohiko Nakamura, Hirokazu Kameoka, Kazuyoshi Yoshii, and Masataka Goto. Timbre replacement of harmonic and drum components for music audio signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7520–7524, Florence, Italy, May 2014.

A Hierarchical Harmonic Mixing Method

Gilberto Bernardes[1(✉)], Matthew E. P. Davies[1], and Carlos Guedes[1,2]

[1] INESC TEC, Sound and Music Computing Group,
Rua Dr. Roberto Frias, 378, 4200 - 465 Porto, Portugal
{gba,mdavies}@inesctec.pt
[2] New York University Abu Dhabi, PO Box 129188, Saadiyat Island,
Abu Dhabi, United Arab Emirates
carlos.guedes@nyu.edu

Abstract. We present a hierarchical harmonic mixing method for assisting users in the process of music mashup creation. Our main contributions are metrics for computing the harmonic compatibility between musical audio tracks at small- and large-scale structural levels, which combine and reassess existing perceptual relatedness (i.e., chroma vector similarity and key affinity) and dissonance-based approaches. Underpinning our harmonic compatibility metrics are harmonic indicators from the perceptually-motivated Tonal Interval Space, which we adapt to describe musical audio. An interactive visualization shows hierarchical harmonic compatibility viewpoints across all tracks in a large musical audio collection. An evaluation of our harmonic mixing method shows our adaption of the Tonal Interval Space robustly describes harmonic attributes of musical instrument sounds irrespective of timbral differences and demonstrates that the harmonic compatibility metrics comply with the principles embodied in Western tonal harmony to a greater extent than previous approaches.

Keywords: Music mashup · Digital DJ interfaces
Audio content analysis · Music information retrieval

1 Introduction

Mashup creation is a music composition practice strongly linked to the various sub-genres of Electronic Dance Music (EDM) and the role of the DJ [27]. It entails the recombination of existing (pre-recorded) musical audio as a means for creative endeavor [27]. As such, it can be seen as a byproduct of existing mass preservation mechanisms and inscribed within the artistic view of the database as a symbol of postmodern culture [20]. Mashup creation is typically confined to technology-fluent composers, as it requires expertise which extends from the understanding of musical structure to the navigation and retrieval of musical audio from large collections. Both industry and academia have been devoting efforts to enhance the experience of digital tools for mashup creation by streamlining the time-consuming search for compatible musical audio.

M. Aramaki et al. (Eds.): CMMR 2017, LNCS 11265, pp. 151–170, 2018.
https://doi.org/10.1007/978-3-030-01692-0_11

Early research on computational mashup creation, focused on rhythmic-only features, particularly those relevant to the temporal alignment of two or more musical tracks [13]. Recent research on this topic has expanded the range of musical attributes under consideration, notably including harmonic-driven features to identify compatible musical audio, commonly referred to as *harmonic mixing*. We can identify three major harmonic mixing methods: key affinity, chroma vectors similarity, and sensory dissonance minimization.

The affinity between musical keys is a prominent method in commercial applications. It is defined by distances across major and minor keys in a double circular representation, known within the DJ community as the *Camelot Wheel*, shown in Fig. 1. This method favors relative major-minor and intervals of fifth relations across musical keys [21] and enforces some degree of tonal stability and large-scale harmonic coherence of the mix by privileging the use of the same diatonic key pitch set. Chroma vector similarity inspects the cosine distance between chroma vector representations of pitch shifted versions of two given audio tracks as a measure of their compatibility [8,9,19]. Distances are typically computed at the beat level, thus privileging small-scale alignments over large-scale harmonic structure between audio slices with highly similar pitch class content. Sensory dissonance models have been used to search for pitch shifted versions of overlapping musical audio which minimize their combined level of roughness [12]; a motivation well rooted in the Western musical tradition by favoring a less dissonant harmonic lexicon.

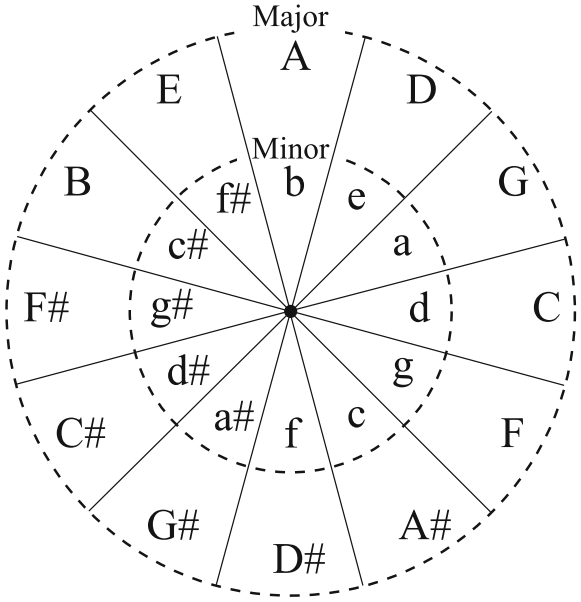


Fig. 1. Key affinity representation based on the two circles of fifths for major and minor keys aligned by relative major-minor key relationships. Enharmonic equivalence is assumed and only sharps (♯) are used.

While existing harmonic compatibility metrics have shown to correlate well with user enjoyment, we argue that they expose promising areas for investi-

gation on harmonic compatibility. First, searching across all possible overlaps between related or in-key (i.e., diatonic) pitch sets result in highly contrasting sonorities with significant levels of enjoyment [3], thus motivating a harmonic compatibility metric below the key level. This metric is also prone to error, namely whenever processing signals with low pitch-to-noise ratio, and despite the perceptual manifestation of the key distances shown in Fig. 1 [17], it denotes temporal phenomena (i.e., key transitions). It remains unclear its validity and usefulness for mixing tracks. Second, while chroma vector distances are effective in capturing highly similar matches between any two given audio tracks, they lack a perceptually-aware basis for comparing pitch configurations [2], and can thus fail to provide an effective ranking between musical audio collections. Third, while psychoacoustic models show enhanced performance over existing approaches, they not only prove to be of limited use when the spectral content of the tracks do not overlap (i.e., when no interaction exists within each critical band), but also violate some perceptual and harmonic principles embodied in Western music, namely at the chordal level, by predicting that an augmented triad is more consonant that a diminished triad [16].

At the design level, existing software for harmonic mixing propose a ranked list of harmonically compatible tracks to a user-defined track [9,21,22]. We believe that this one-to-many mapping is reductive in offering a global view of a music collection and enabling a fluid navigation through an audio collection. Furthermore, it is computationally inefficient, as it recomputes highly intensive audio signal analysis every time a different audio track is selected as target.

In light of these limitations, we propose a new method for computing the small- and large-scale harmonic compatibility between a beat-matched collection of audio tracks, based on indicators from the perceptually-motivated Tonal Interval Space [2] and following the diagram architecture shown in Fig. 2. The proposed method has three aims: (i) to perceptually enhance the manifestation of metrics for harmonic compatibility, (ii) to inspect small- and large-scale structural levels by summarizing existing mashup creation approaches in a single framework, and (iii) to efficiently explore musical audio collections, without the need for intensive computation for each specific target, towards a more fluid user-experience which fosters experimentation.

The remainder of this paper is structured as follows. Section 2 reviews the Tonal Interval Space, which we adapt towards an enhanced representation of the harmonic content of musical audio. Section 3 presents content-driven harmonic analysis of musical audio. Section 4 introduces new metrics for computing the harmonic compatibility between audio tracks. Section 5 details an interactive visualization which exposes the compatibility of a musical audio collection. Section 6 presents an evaluation of the indicators and compatibility metrics which underpin our harmonic mixing method. Finally, Sect. 7 presents conclusions and areas for future work.

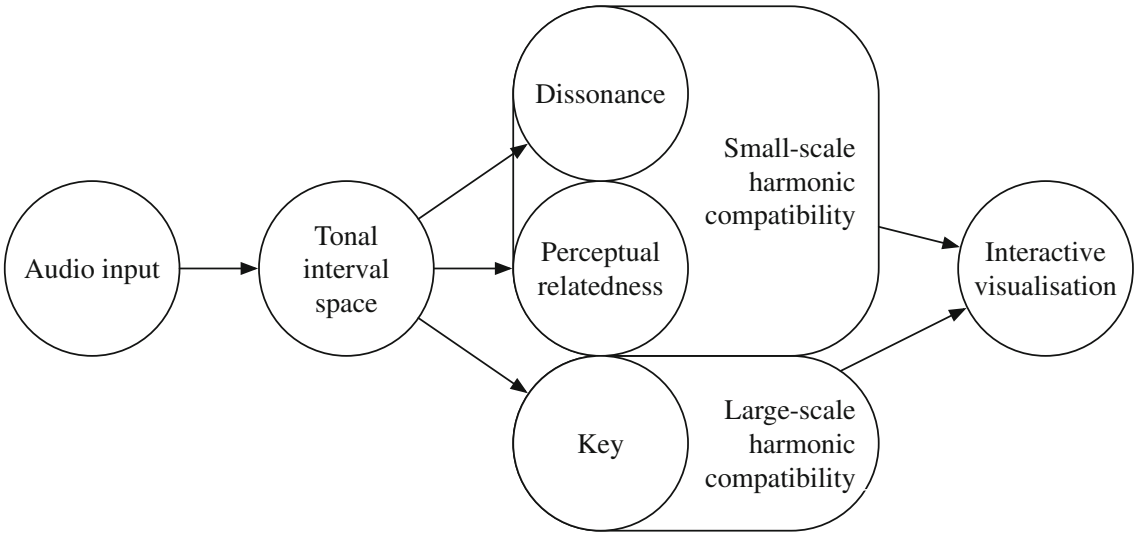


Fig. 2. Diagram of the component modules of our compatibility method for music mixing.

2 Adapting Tonal Interval Vectors for Musical Audio

We represent the harmonic content of musical audio tracks as 12-dimensional Tonal Interval Vectors (TIVs) [2]. This vector space creates an extended representation of tonal pitch in the context of the *Tonnetz* [11], named the Tonal Interval Space, where the most salient pitch levels of tonal Western music—pitch, chord, and key—exist as unique locations. TIVs, $T(k)$, are computed from an audio signal as the weighted Discrete Fourier Transform (DFT) of an L_1 normalized chroma vector, $c(n)$, such that:

$$T(k) = w_a(k) \sum_{n=0}^{N-1} \bar{c}(n) e^{\frac{-j2\pi kn}{N}}, \quad k \in \mathbb{Z}. \tag{1}$$

where $N = 12$ is the dimension of the chroma vector, each of which expresses the energy of the 12 pitch classes, and $w_a(k)$ are weights derived from empirical ratings of dyads consonance used to adjust the contribution of each dimension k (or interpreted musical interval) of the space, which we detail over the next paragraphs. We set k to $1 \leq k \leq 6$ for $T(k)$ since the remaining coefficients are symmetric. $T(k)$ uses $\bar{c}(n)$ which is $c(n)$ normalized by the DC component $T(0) = \sum_{n=0}^{N-1} c(n)$ to allow the representation and comparison of music at different hierarchical levels of tonal pitch [2]. To represent variable-length audio tracks, we accumulate chroma vectors, $c(n)$, resulting from 16384 sample windows analysis at 44.1 kHz sampling rate ($\approx$ 372 ms) with 50% overlap across the track duration.

In [2], we used two complementary sources of empirical data—empirical ratings of dyad consonance shown in Table 1 [14] and the ranking order of triad consonance: {maj, min, sus4, dim, aug} [1,7]—to make the Tonal Interval Space perceptually relevant for symbolic music input representations (i.e., binary chroma vectors). Here, we revisit the task to comply with the timbral components of

Table 1. Composite consonance ratings of dyads consonance [14].

Interval class	m2/M7	M2/m7	m3/M6	M3/m6	P4/P5 TT
Consonance	–1.428	–.582	.594	.386	1.240 –.453

musical audio. Our goal is to find a set of weights, $w_a(k)$, which regulate the importance of the DFT coefficients k in Eq. 1, so that the space conveys a reliable consonance indicator correlated with the aforementioned empirical ratings of dyad [14] and triad consonance [7]. The applied method follows the previously used brute force approach [2], which produced a near optimal result.

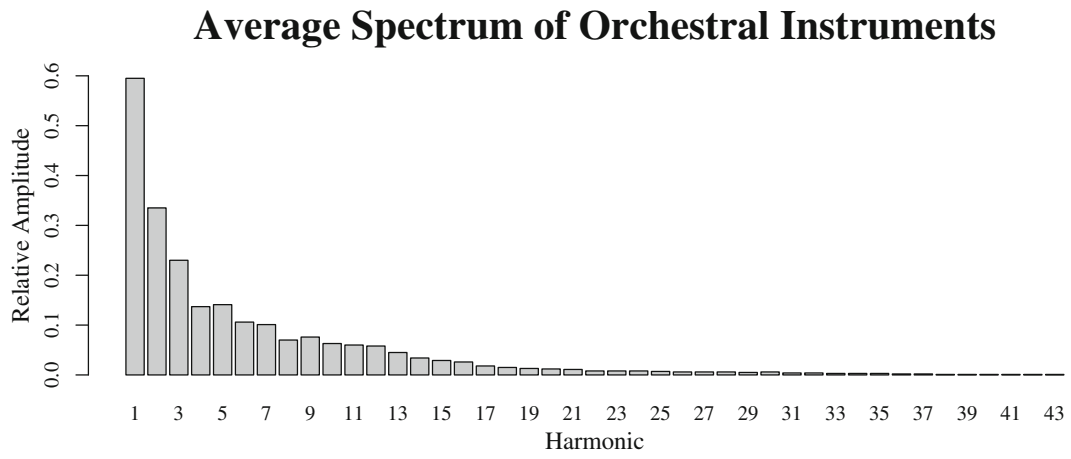


Fig. 3. Average harmonic spectrum of 1338 tones from orchestral instruments [23].

A major problem in defining a set of weights for robustly representing musical audio in the Tonal Interval Space is the variability of timbre across musical instruments and registers. A refined model capable of tracing the idiosyncratic timbral attributes of a particular instrument raises scalability and complexity issues which would defeat the value of the Tonal Interval Space in providing effective and, most importantly, efficient perceptual indicators of tonal pitch. To circumvent these issues, we adopt the 43-partial harmonic spectrum template shown in Fig. 3 to represent the harmonic content of musical audio. The template results from averaging 1338 recorded instrument tones from 23 Western orchestral instruments and can be understood as a time-invariant spectrum of an "average instrument" [23].

To allow a computationally tractable search for weights to represent musical audio in the Tonal Interval Space, we split the task into two steps. In the first step, we find the weights, $w_a(k)$, from all possible 6-element combinations (with repetition and order relevance), of the set $I = \{1, 19\} \in \mathbb{Z} : I = 2I + 1$ (a total of approximately one million combinations), which maintain in the Tonal Interval Space the empirical ranking order of common triads consonance [7]. Following [2], we compute the consonance of musical audio triads in the Tonal Interval Space

as the norm of TIVs, $\|T(k)\|$, which we detail in Sect. 3. In the second step, from the resulting set of 111 weight vectors which preserve the ranking order of empirical triads consonance (i.e., a Spearman rank correlation $\rho = 1$), we identify those which have the highest linear correlation to the empirical dyad consonance ratings shown in Table 1.

We repeated the two aforementioned steps to further optimize the two weight vectors ($\{1, 7, 15, 11, 13, 7\}$ and $\{3, 7, 15, 11, 13, 7\}$) with the highest linear correlation ($r = .988$), below our minimal interval using 0.5 increments.

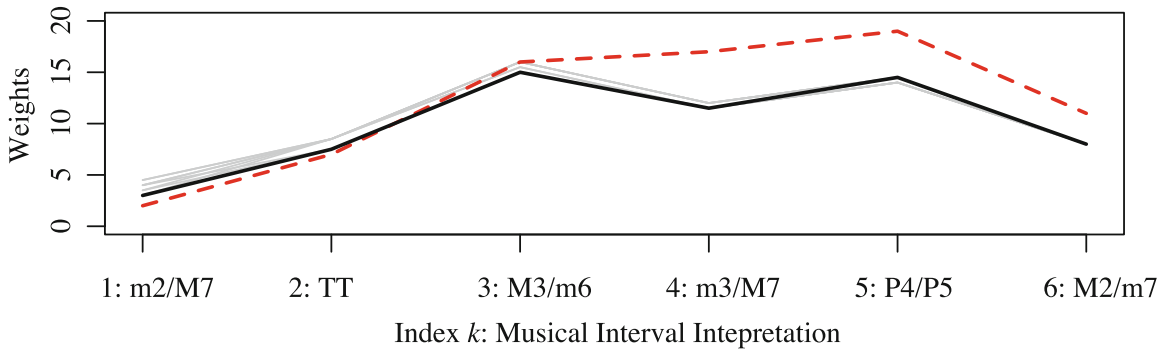


Fig. 4. The set of weights that maximize the linear ($r > 0.995$) and ranking order ($\rho = 1$) correlation of Tonal Interval Space's musical audio consonance indicator with empirical ratings of dyad and triad consonance, respectively. The bold line corresponds to the set of weights $w_a(k)$ used in Eq. 1 and the dashed line to the weights, $w_s(k)$, defined in [2] for a symbolic based Tonal Interval Space.

Figure 4 shows 11 sets of weights, $w_a(k)$, which preserve empirical triads ($\rho = 1$) and dyad ($r > .99$) consonance for musical audio. Given the inherent similarity in shape of the different sets of weights and their almost perfect linear relationship, we do not believe the choice over exactly which set of weights to be critical. Ultimately, we selected the weights with the greatest mutual separation between the triads according to consonance, thus $w_a(k) = \{3, 8, 11.5, 15, 14.5, 7.5\}$.

3 Harmonic Indicators from Musical Audio: Dissonance and Perceptual Relatedness

To provide a mathematical representation of Western tonal harmony perception as distances in the Tonal Interval Space, we distorted a DFT space according to the weights, $w_a(k)$, derived from empirical consonance ratings. In light of this design feature and following previous metrics detailed in [2], we can compute two indicators of musical audio dissonance, D, and perceptual relatedness, R, from the space as distance metrics.

$$D = 1 - \left(\frac{\|T(k)\|}{\|w_a(k)\|} \right), \tag{2}$$

$$D_{ij} = 1 - \left(\frac{\|a_i T_i(k) \ + a_j T_j(k)\|}{a_i + a_j \|w_a(k)\|} \right), \tag{3}$$

and

$$R_{i,j} = \sqrt{\sum_{k=1}^{M} |T_i(k) - T_j(k)|^2}. \tag{4}$$

We adapt the consonance metric presented in [2] to a musical audio dissonance metric, by subtracting the normalized norm of a TIV, $T(k)$, from one. Drawn from the properties of the DFT at the basis of the space, the location of multi-pitch TIVs is equal to the linear combination of its component pitch classes [2]. Thus, we can efficiently compute the dissonance of two combined TIVs, $T_i(k)$ and $T_j(k)$, representing two overlapping audio tracks, i and j, using Eq. 3, where a_i and a_j are the amplitudes of $T_i(k)$ and $T_j(k)$.

Equation 4 computes the perceptual relatedness, $R_{i,j}$, as the Euclidean distance between TIVs. Small values of perceptual relatedness, R, denote voice leading parsimony and controlled transitions of the interval content between neighborhood TIVs, as smaller distances primarily enforce the number of shared tones, and to a lesser degree, the interval relations imposed by the weights, $w_a(k)$.

4 Harmonic Compatibility Metrics

Based on the two dissonance, D, and perceptual relatedness, R, indicators from the Tonal Interval Space presented in Sect. 3, we now propose two metrics that aim at capturing the harmonic compatibility between TIVs to be mixed. Of note is the split between small- and large-scale harmonic compatibility, which roughly correspond to the 'sound object' and 'meso' or 'macro' time scales of music, respectively. In other words, the small-scale denotes the basic units of musical structure, from notes to beats, and the large-scale inspects the structural levels between the phrase and the overall musical piece architecture [24]. In the context of our work, the first aims mostly at finding good harmonic matches between the tracks in a collection, and the second in guaranteeing control over the overall harmonic structure of a mix, i.e., the tonal changes at the key level across its temporal dimension.

4.1 Small-Scale Harmonic Compatibility

The level of small-scale harmonic compatibility is expressed as the combination of two harmonic audio indicators from the Tonal Interval Space detailed in Sect. 3: dissonance, D, and perceptual relatedness, R. The latter indicator finds sonorities which have a strong perceptual affinity and thus range from a perfect match to sonorities with different timbres and similar pitch content, to an array of sonorities with increased levels of perceptual distance. We envisage it as an

extension of the chroma vector similarity method used as a measure of harmonic compatibility in prior studies [8,9,19], which offers a refined control over the introduction of new tones as well as its interval relations in the resulting mix between overlapped tracks.

Given the likely increase in dissonance in the mix and following some perceptual evidence from previous research [12], our small-scale harmonic compatibility also privileges the search for less dissonance mixes—a well established principle in the common syntax of Western tonal harmony [1,5,18]. Hence, our small-scale harmonic compatibility metric, H, is then computed as the product of the two indicators, such that:

$$H_{i,j} = \bar{R}_{i,j} \cdot \bar{D}_{ij}, \tag{5}$$

where $\bar{R}$ and $\bar{C}$ are R and C scaled to the range $\{0, 1\} \in \mathbb{R}$ to balance the importance of both indicators in the compatibility metric. The main motivation for the simple multiplication of the two variables is rooted in the visualization method we detail in Sect. 5, notably by enforcing a small-scale harmonic compatibility, $H = 0$, when comparing the same track.

4.2 Large-scale Harmonic Compatibility

A derivation of the perceptual relatedness indicator, R, exposes an important property of the Tonal Interval Space: the formation of fuzzy key clusters of diatonic pitch class sets. Neighborhood relations between these clusters in the Tonal Interval Space result in a representation similar to Fig. 1, as a result of common-tone relations between keys. This property is adopted to estimate the global key from musical audio track, which aims to guide users in planning the large-scale harmonic structure of a mix.

We use the method reported in [4] to compute the global key estimate, Q, of an audio track in the Tonal Interval Space, as the minimum Euclidean distance of an audio input TIV, $T(k)$, from the 12 major and 12 minor key TIVs, $T_r(k)$, such that:

$$Q = \text{argmin}_p \sqrt{\sum_{k=1}^{6} |T(k) \cdot \alpha - T_r(k)|^2}, \tag{6}$$

where $T_r(k)$ is derived from a collection of templates (understood here as chroma vectors) representing pitch class distributions for each of the 12 major and 12 minor keys [26]. When $r \leqslant 11$, we adopt the major profile and when $r \geqslant 12$, the minor profile. $\alpha = 0.35$ is a factor which displaces input sample TIVs to balance predictions across modes [4] (Table 2).

The estimated key, Q, ranges between $0 - 11$ for major keys and $12 - 23$ for minor keys, where 0 corresponds to C major, 1 to C# major, and so on through to 23 being B minor.

5 Interactive Visualization

We created a software prototype in Pure Data which implements the proposed hierarchical mixing method, notably the harmonic compatibility metrics. In light

Table 2. Sha'ath's [26] key profiles, p, for the C major and C minor keys.

Key	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
C major	7.239	3.504	3.584	2.845	5.819	4.559	2.448	6.995	3.391	4.556	4.074	4.459
C minor	7.003	3.144	4.359	5.404	3.672	4.089	3.907	6.200	3.634	2.872	5.355	3.832

of the possibility to retrieve compatible harmonic tracks from large musical audio collections, we designed an interactive visualization which aims to (i) provide a global view of the harmonic compatibility across all audio tracks in a collection (i.e., many-to-many relationships); (ii) expose a hierarchical representation over the harmonic compatibility between tracks; and (iii) promote user experimentation and creative endeavorer.

To this end, we pursued an interface design based on crossmodal associations between sound and image, for which a screenshot is shown in Fig. 5. All audio tracks in a collection are represented graphically in a two-dimensional (2-D) space, where regular polygons denote audio tracks and grey circles key centers. Distances among polygons (i.e., audio tracks) indicate small-scale harmonic compatibility, H, and links from circles (i.e., key centers) to polygons the large-scale compatibility, Q, or, in other words, the association to its estimated key.

The computation of 2-D coordinates for each audio track from a square matrix of all pairwise tracks small-scale harmonic compatibility distances, H, is a classical problem, which can be solved by a specific class of algorithms, notably including Multidimensional Scaling (MDS). From m musical audio tracks in a collection, we compute an $H_m \cdot H_m$ harmonic compatibility square matrix, from which an MDS representation extracts two-dimensional coordinates for each sample. The resulting representation attempts to preserve the inter-sample small-scale harmonic compatibility with minimal distortion.

The computation of coordinates for each key center equals the convex combination (or the centroid, in geometric terms) of the 2-D coordinates of its estimated tracks. We adopted this efficient method for the computation of key centers based on the theoretical assumption that the diatonic set of a key is denoted in the Tonal Interval Space by the convex combination of set of diatonic note TIVs [2]. Therefore, assuming that all tracks with the same key estimate represent well its diatonic set, its corresponding key coordinates ought to be represented with minimal distortion.

Two additional rhythmic and spectral musical features of each track are represented by graphical attributes of the polygons (number of sides and color, respectively) to expand the search attributes to fit particular compositional goals. The number of sides, ranging from three to six, expose the note onset density, computed by a threefold approach. First, we extract a spectral flux onset detection function, from a windowed power spectrum representation of the audio signal (2048 analysis windows size at 44.1 kHz sampling rate with 50% overlap), using the `timbreID` [6] library within Pure Data. Second, we identify the peaks from the function above a user-defined threshold, t, whose temporal location

we assume to indicate note onset times. Prior to the peak detection stage, we apply a bi-directional low-pass IIR filter, with a cutoff frequency of 5 Hz to avoid spurious detections. Finally, we compute the ratio between the number of onsets and the entire duration of the audio file in seconds and scale the values for a given audio collection to the $\{3, 6\} \in \mathbb{Z}$ range of polygon sides.

The polygons' color, ranging from continuous shades of yellow to red, represents the spectral region a sample occupies in the perceptual perceptually-motivated Bark frequency scale.[1] A threefold strategy is adopted to map these two dimensions. First, we accumulate Bark spectrum B_b, representations computed on short-time windows of 2048 samples size at 44.1 kHz sampling rate with 50% overlap across an audio track, again using the `timbreID` [6] library within Pure Data. Then, we extract the centroid as an indicator of its spectral region, S, using Eq. 7. Finally, we map the spectral region, S, value to the the color scheme. Bark band 1 corresponds to yellow, and bark band 24 to red. Between these values, the colors are linearly mixed.

$$S = \frac{\sum_{i=1}^{19} B_b \cdot b}{\sum_{i=1}^{19} B_b}, \tag{7}$$

where B_b is the energy of the bark band b. The S indicator can range from 1 to 24.

The user can interact with the visualization by clicking on the polygons to trigger their playback, thus promoting an intuitive search for compatible tracks as well as strategies for serendipity and experimentation, rather than a fully automatic method for mashup creation. A demo of this interactive visualization can be found online at: https://sites.google.com/site/tonalintervalspace/mixmash.

6 Evaluation

We undertake a twofold strategy to evaluate our harmonic mixing method. First, we assess the perceptual validity and degree of timbre invariance of the two dissonance, D and perceptual relatedness, R, indicators from the Tonal Interval Space, which underpin our small-scale harmonic compatibility metric. Particular emphasis is given to the implications of the newly proposed weights, $w_a(k)$, for representing musical audio. Second, we examine the level of compliance of the proposed harmonic compatibility and related metrics with Western tonal music principles.

Unless otherwise specified, across evaluation tasks the harmonic spectrum of musical audio is computed as the sum of individual notes spectra using the harmonic template of an average instrument shown in Fig. 3.

[1] The Bark spectrum balances the resolution across the human hearing range in comparison to the typical power spectrum representation, namely increasing the resolution in the low frequency region. It is computed by warping a power spectrum to the 24 critical bands of the human auditory system [28].

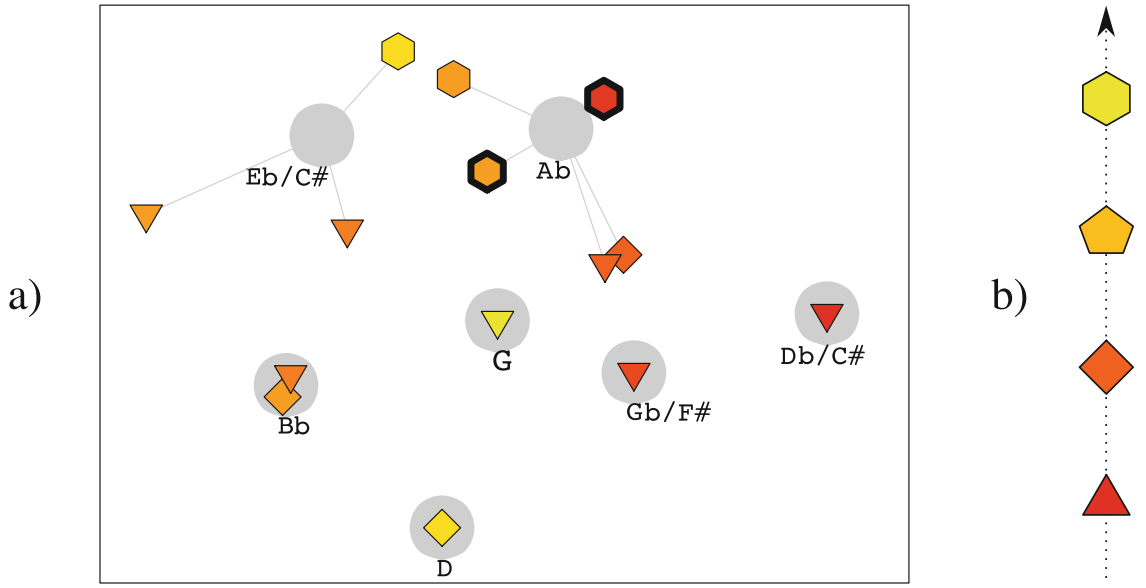


Fig. 5. (a) Interactive visualization of the hierarchical harmonic compatibility between all audio tracks in a collection. Polygons represent audio tracks and circles key centers. Polygon distances indicate small-scale harmonic compatibility and the links from circles the large-scale harmonic compatibility. The graphical attributes of the polygons show onset density (number of sides) and spectral region (color). Polygons with thick outlines indicate the selected files currently playing. (b) The ranking order of low to high onset density and spectral region.

The Spearman rank correlation, ρ, is the metric used to compare most data in our evaluation. It measures the strength and direction of the monotonic relationship between two variables. The motivation to adopt such a metric is due to the importance of the ranking order in proposing harmonic mixes rather than ensuring a linear relationship between the variables (computed, for example, by the Pearson correlation) and the prevalence of ranked data in perceptual studies. The result of the Spearman rank correlation, ρ, is expressed by a single correlation coefficient value in the $\{-1, +1\} \in \mathbb{R}$ range. Positive and negative correlation coefficients express positive and negative relationships between variables, respectively, and a correlation coefficient of $\rho = 0$ indicates that no relationship between the variables exists.

6.1 Harmonic Indicators for Musical Audio

We assess how the weights, $w_a(k)$, implemented as a design feature of the space, to provide a dissonance, D, indicator of musical audio, compare to (i) the sensory dissonance metric by Hutchinson and Knopoff [15] at the basis of Gebhardt et al. [12] mashup creation system and (ii) the previously proposed weights, $w_s(k) = \{2, 11, 17, 16, 19, 7\}$, adjusted for symbolic music representations [2], in measuring triads dissonance—the chordal level at which most mashup creation exists. All theoretical models are additionally compared to perceptual dissonance data [1,7].

Then, we assess how the perceptual relatedness, R, metric and the cosine similarity between chroma vectors, adopted from Davies et al. [9], as a harmonic compatibility metric, compare to perceptual data [25]. The dyad pitch level is used to undertake this comparison as it lays out the basis for distances at all higher hierarchical pitch levels. Furthermore, various corroborating perceptual studies exist for this pitch level, which Schwartz et al. summarize in [25].

Finally, we determine the degree of timbre invariance of both Tonal Interval Space indicators across a wide range of musical instrument timbres. To this end, we extend the two previous tasks beyond the theoretical levels by evaluating the indicators across multiple musical instruments and registers. To constrain the experiment, we limited the pitch sets to triads in the root position with stacked thirds and dyads no larger than one octave.

The pitch sets result from the sum of individual note recordings from acoustic and electronic instruments. IRCAM's Studio OnLine (SOL) database[2] is adopted for the acoustic instruments and the NSynth database [10] for the electronic instruments. From the entire collection of acoustic instruments in the SOL database, we selected four instruments from each family, aiming to cover a wide note range: strings (violin, viola, guitar, and violoncello), woodwinds (flute, B♭ clarinet, alto saxophone, and bassoon), and brass (trumpet, trombone, horn, and bass tuba). From the NSynth database, we selected four electronic and synthetic instruments which commonly feature in EDM: electric keyboard, synth lead, and electric bass, and electric guitar.

The selected acoustic instrument samples are quasi-stationary, i.e., without any extended playing technique, and electronic and synthetic instrument samples are non-stationary, with clear temporal changes at a regular fast pace, as a result of audio effects such as tremolo, vibrato, and filtering.[3] A *mezzo-forte* dynamic was adopted in both cases. Besides the alignment of the samples on detected onsets, no further processing was applied. Due to some discrepancies in the duration of the instrument samples in both databases, we limited their duration to two seconds, thus ensuring the same duration across all pitch sets. Instrument samples are mono WAV files with 44.1 Khz sampling rate and 16 bit depth. We computed the indicators (i.e., the dissonance, D, of triads and the perceptual relatedness, R, of dyads) per instrument as the average value of overlapping 8192 sample windows at 44.1 kHz sample rate with 50% overlap.

6.2 Harmonic Compatibility Metrics

We assess the extent to which the principles embodied in Western tonal harmony, namely the prevalence of common chord sets with reduced dissonance, are promoted by our proposed small-scale harmonic compatibility metric, H, and

[2] We used the version 0.9 of IRCAM's SOL database, retrieved at http://forumnet.ircam.fr/product/orchids-en/ in July, 2017 as the supporting database of the Orchids software.

[3] Please refer to https://sites.google.com/site/tonalintervalspace/mixmash to listen to electronic and synthetic instrument sample examples from the NSynth database.

related approaches, namely chroma similarity [9] and sensory dissonance [12]. To this end, we inspect which pitch class sets are identified as most compatible from a total of 55 triads, which result from overlapping the pitch class C or 0 (i.e., index 0 in $c(n)$ from Eq. 1) to all remaining pitch class dyads, in each metric. All dyads resulting from the combination (without repetition and order relevance) of the $\{1-11\} \in \mathbb{Z}$ set are considered.

6.3 Results

Table 3 reports the perceived and computed dissonance level of common musical audio triads. The perceptual data have been corroborated by several experimental studies [1,7]. The values for sensory dissonance are taken from [15] and result from applying the metric to triads that lie within the $\mathtt{C}^4$-$\mathtt{C}^5$ octave. The reported Spearman rank correlations, ρ, and their significance values, p, between the perceptual data and theoretical models show that the Tonal Interval Space is more consistent in ranking the dissonance of common triads than the Hutchinson and Knopoff [15] sensory dissonance model used in related mashup literature [12]. Moreover, we demonstrate that the weights, $w_a(k)$, computed in Sect. 2 are decisive in capturing the dissonance of musical audio triads in the Tonal Interval Space, as the previously proposed weights, $w_s(k)$ for symbolic music inputs [2] fail at providing a ranking of triads consonance from musical audio in line with perceptual data.

Table 3. Ranking of triads dissonance from perceptual data [1,7] and two theoretical models: sensory dissonance [15] and the Tonal Interval Space dissonance, adopting two sets of weights adapted to symbolic representation, $w_s(k)$, and musical audio, $w_a(k)$. The Spearman rank correlations, ρ, and their significance values, p, between the perceptual data and theoretical models are reported.

Triad quality	Perceptual rank [1,7]	Sensory dissonance [15]	Tonal Interval Space (D)	
major	1	1 (.139)	1–2 (.768)	1–2 (.783)
minor	2	2 (.148)	1–2 (.768)	1–2 (.783)
sus4	3	4 (.228)	3 (.769)	3 (.784)
dim	4	5 (.230)	5 (.819)	4 (.805)
aug	5	3 (.149)	4 (.780)	5 (.806)
Correlation ρ		.700	.878	.975
Significance p		.233	.054	$<.05$

Table 4 reports the perceived and computed dyads relatedness, R. The perceptual data are taken from [25], which summarizes different experimental studies. The chroma similarity was computed as the cosine similarity between chroma vectors following [9]. The reported Spearman rank correlations, ρ, and their significance values, p, between the perceptual data and theoretical models show

that the Tonal Interval Space is more consistent in ranking the perceptual relatedness of dyads than the chroma similarity. Moreover, the ranking order of dyad relatedness in the Tonal Interval Space is consistent with tonal harmony principles in the sense it promotes tertian harmony as a result of having fifths and thirds at a closer distance than all remaining intervals. The chroma similarity, adopted as a harmonic compatibility metric in previous computational mashup works [8,9,19], largely agrees with the dyad perceptual ranking, with the notable exception of the minor seconds and major seventh which are closer in this metric space than the major and minor thirds or their complementary minor and major sixth, thus disrupting a preference for tertian harmonies.

Table 4. Ranking of dyad relatedness from perceptual data and theoretical models. The Spearman rank correlations, ρ, and their significance values, p, between the perceptual data and theoretical models are reported.

Dyad	Perceptual rank [25]	Chroma similarity [9]	Tonal Interval Space (R)
Unison (P1)	1	1−2 (0.00)	1−2 (0.00)
Octave (P12)	2	1−2 (0.00)	1−2 (0.00)
Perfect fifth (P5)	3	3−4 (1.11)	3−4 (1.37)
Perfect fourth (P4)	4	3−4 (1.11)	3−4 (1.37)
Major third (M3)	5	7−8 (1.20)	7−8 (1.62)
Major sixth (M6)	6	10−11 (1.26)	5−6 (1.53)
Minor sixth (m6)	7	7−8 (1.20)	7−8 (1.62)
Major third (M3)	8	10−11 (1.26)	5−6 (1.53)
Tritone (TT)	9	9 (1.25)	9 (1.78)
Minor seventh (m7)	10	12−13 (1.27)	10−11 (1.79)
Major second (M2)	11	12−13 (1.27)	10−11 (1.79)
Major seventh (M7)	12	5−6 (1.16)	12−13 (1.95)
Minor second (m2)	13	5−6 (1.16)	12−13 (1.95)
Correlation ρ		.609	.956
Significance p		$< .05$	$< .001$

Table 5 shows the Spearman rank correlations between perceptual data [1, 15,25] and Tonal Interval Space indicators from musical instrument inputs. We inspected the dissonance, D, of common triads and the perceptual relatedness, R of dyads. With the sole exception of the electric bass, all instruments have a significant Spearman rank correlation between the perceptual data and their computed indicators, thus ensuring a high degree of timbral invariance in computing the harmornic indicators from the Tonal Interval Space.

These results should also be read in light of the Spearman correlation between theoretical musical audio representations and perceptual data shown in Tables 3 and 4. The triad dissonance that results from instrument recordings does not fully mirror the perfect monotonic relationship of the theoretical results. Looking across instrument families it is noticeable the optimal results across all inspected

Table 5. Spearman rank correlation, ρ, of perceptual data for triads consonance and dyads distances and dissonance, D, and perceptual relatedness, R, metrics from the Tonal Interval Space, respectively, across multiple instruments. All results are significant for $p < 0.05$, except for the electric bass dissonance, where $p = 0.233$.

	Intrument	Pitch range (MIDI)	Dissonance (D)	Perceptual relatedness (R)
Strings	Violin	55−100	1	.956
	Viola	48−96	.9	.973
	Guitar	38−83	.8	.896
	Violoncello	36−84	.9	.973
Woodwinds	Flute	59−96	.9	.945
	B♭ clarinet	50−91	1	.956
	Alto saxophone	49−81	.9	.940
	Bassoon	34−75	1	.934
Brass	Trumpet	54−86	1	.951
	Trombone	34−72	1	.951
	Horn	31−77	1	.956
	Bass tuba	30−65	1	.945
Electronic	Electric guitar	36−86	1	.951
	Synth lead	21−108	.9	.934
	Electric keyboard	21−108	.9	.912
	Electric bass	9−96	.7	.900

instruments in the brass family. The remaining families have the approximately same number of instruments which do not fully comply with the perceptual triad ranking. The small sample of observed instruments raises interesting issues which should ultimately be addressed in future adaptations of the Tonal Interval Space. The dyad perceptual relatedness that results from instrument recordings is in line with the theoretical results ($\rho = 956$).

Figure 6 shows the analysis of the electronic instruments, which the "average (acoustic) instrument" spectrum, at the basis of the adaptation of the Tonal Interval Space to musical audio, does not model. In both the triad dissonance and dyad perceptual relatedness plots, an monotonically increasing function is expected, given the order of the x-axis elements according to an ascending perceptual ranking. In the triad dissonance plot, we can observe that the perceptual ranking is not preserved by violating the order of different triads per instrument. The dyad perceptual relatedness plot questions the symmetric property of the space (as a result of the DFT) for complementary intervals (e.g., m2 and M7 or M2 and m7) whose dissonance levels are averaged, thus neglecting any distinction between them.

Figure 7 shows the seven best ranking triads resulting from the overlap of the pitch class `C` and all remaining 55 pitch class dyad combinations for the following five metrics: (i) sensory dissonance [15]; (ii) chroma similarity [9]; (iii) perceptual relatedness, R; (iv) dissonance, D; and (v) small-scale harmonic compatibility,

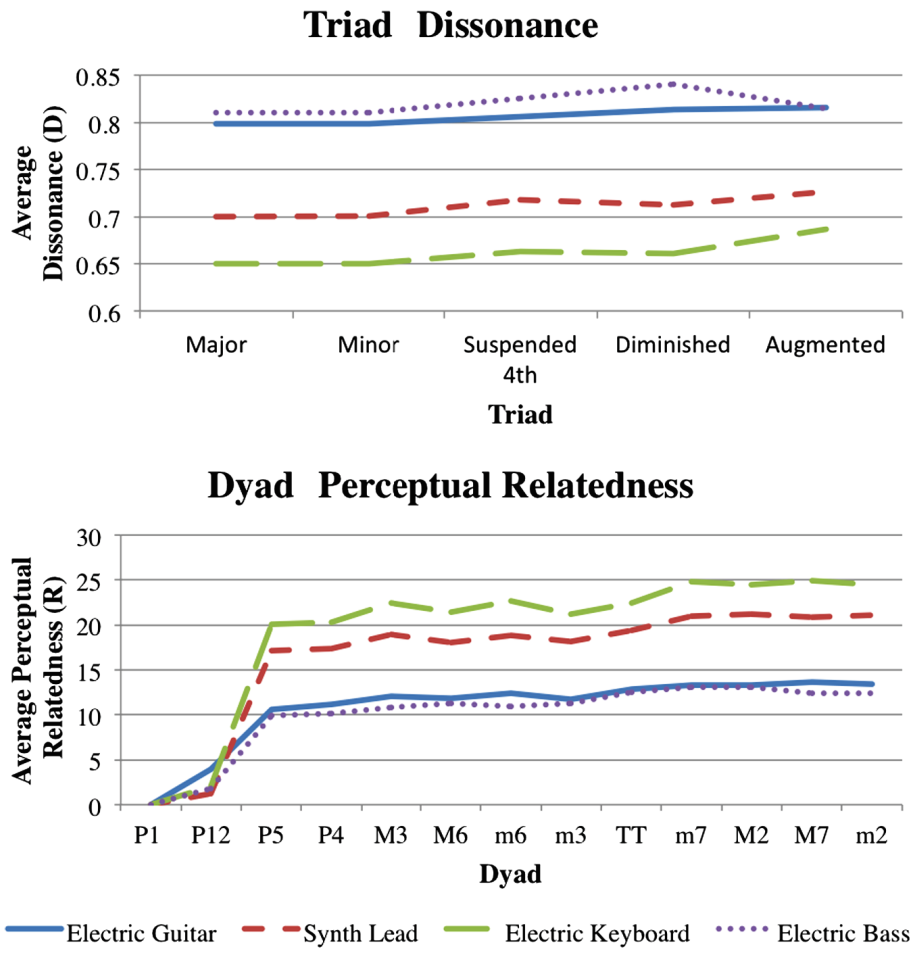


Fig. 6. Computed dyad dissonance and triad perceptual relatedness of electronic and synthetic instrument samples from the Tonal Interval Space. Pitch sets result from summing individual instrument samples. Average values across the inspected instruments' pitch range are reported for both indicators.

H. Sensory dissonance ranking order favors triads in line with the results presented in Table 3. Besides the preference for augmented over suspended 4ths and a few minor triads, to a large extent it conveys the expectancy of the Western tonal syntax. The ranking order of triads in the chroma similarity space and the perceptual relatedness, R, are aligned with the findings shown in Table 4. The chroma similarity favors chords including P4/P5 and m2/M7 intervals. Conversely, the perceptual relatedness, R, favors chords including P5/P4, m3/M6, and M3/m7 intervals. As such, combining sonorities with small R values, results in extended chords with stacked fifths and thirds. However, while the chords resulting from these vertical aggregates are building blocks of Western tonal harmony, the best ranked chords result in multiple seventh chords (with omitted notes), and not, as expected in the ideal case, triads (e.g., major, minor, and diminished).

When combining the perceptual relatedness, R, and the dissonance, D, indicators, i.e., when adopting our small-scale harmonic compatibility, H, metric, we enforce the preference for common major, minor, and suspended fourth

triads—the most common building blocks of the Western tonal harmony, by favoring in the former ranking less dissonant triads. Nonetheless, the small-scale harmonic compatibility, H, ranking in Fig. 7 ignores the key level, thus promoting chromaticism (non-diatonic) progressions between neighbor sonorities. Our large-scale harmonic compatibility addresses this issue by providing in our interactive visualization a layer of information which can guide users in selecting (in-key) diatonic mixes.

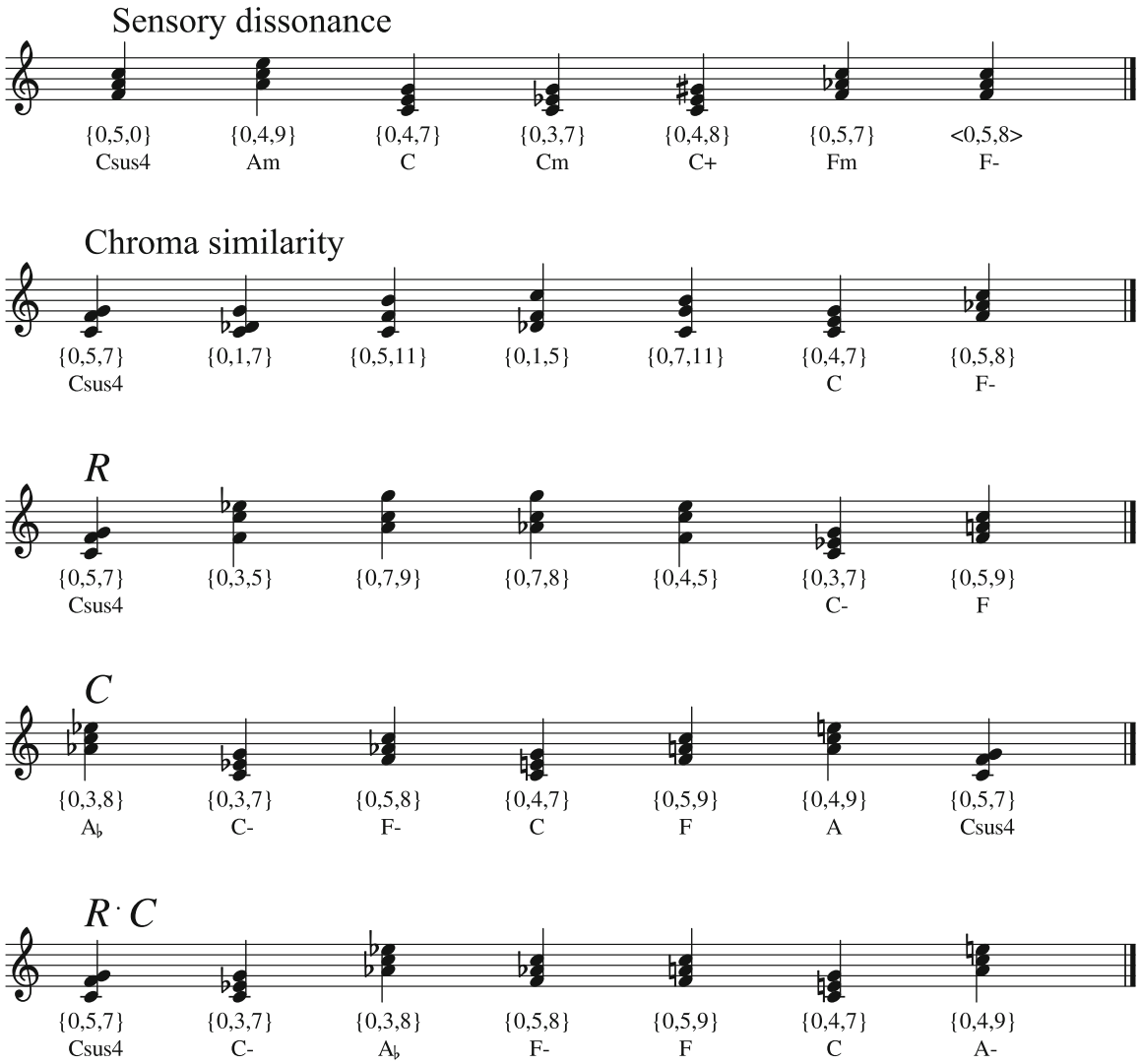


Fig. 7. Ranking order of triads resulting from overlapping the pitch class `C` and all remaining pitch class dyads as given by computed metrics. Apart from triad sensory dissonance, which is taken from [15], the remaining models were computed using the harmonic spectrum representation of an "average instrument" spectrum shown in Fig. 3. To the pitch class set of the resulting triads, we include the chord label whenever unambiguous and complete triads are formed. Sounding examples of the table contents are available at: https://sites.google.com/site/tonalintervalspace/mixmash.

7 Conclusion and Future Work

In this paper we have presented a hierarchical harmonic mixing method with two underlying metrics that inspect the harmonic compatibility of musical audio tracks at both small- and large-scale structural levels. Small-scale harmonic compatibility results from the combination of dissonance and perceptual relatedness indicators from the Tonal Interval Space, which we adapted to represent musical audio. Our adaptation is largely invariant to timbral differences of instrument sounds, and aims to assist users in finding good local alignments between mixed tracks. Large-scale harmonic compatibility relies on key estimates and aims to assist users in planning the global harmonic structure of a mix. A software prototype in Pure Data presents the metrics to the user in an interactive visualization. Crossmodal associations between sound attributes and geometric elements aim at promoting a global exploration of an audio collection, namely by fostering a fluid strategy to retrieve harmonically compatible tracks.

In future work, we plan to address three important issues raised by the current evaluation. The first is related to the perceptual validity of the harmonic compatibility metrics. Despite the perceptual motivation of its indicators, their combination as a result of a simple multiplicative fashion, remains open. Our evaluation shows that the model leads to an attractive theoretical result, but we speculate that a better perceptually-grounded quantification can be found.

The second issue is related to the relevancy of the proposed method for EDM and the creative flow of the DJ in meaningful world-case application scenarios. One can argue whether the design of the Tonal Interval Space design oriented towards Western tonal music harmony is a prominent dimension in EDM. This ought to be investigated by a broader study which not only takes into account the various dimensions of musical structure at hand, but also the user experience promoted by our interface in the context of EDM practice. In response, we can state that the interface design reflects these concerns as it guides the users through the creative process based on metrics aligned with some perceptual findings rather then dictating or automating the process based on some judgment about the harmonic quality of the music. To this end, we believe that a large degree of creative endeavor can be achieved.

The third aspect under consideration in future work relates to the scalability of the data under analysis. The current interactive interface is inefficient when we scale the musical collection above a certain number of tracks, as it results in dense cluttered visual clusters. Given the goal of inspecting large musical audio collections at the level of the hundreds or thousand tracks, we aim to address strategies to enhance the sparseness of the visualization.

Acknowledgments. This work is supported by national funds through the FCT - Foundation for Science and Technology, I.P., under the project IF/01566/2015.

References

1. Arthurs, Y., Beeston, A.V., Timmers, R.: Perception of isolated chords: Examining frequency of occurrence, instrumental timbre, acoustic descriptors and musical training. Psychol. Music **46**(5), 662–681 (2018). https://doi.org/10.1177/0305735617720834
2. Bernardes, G., Cocharro, D., Caetano, M., Guedes, C., Davies, M.: A multi-level tonal interval space for modelling pitch relatedness and musical consonance. J. New Music Res. **45**(4), 281–294 (2016)
3. Bernardes, G., Cocharro, D., Guedes, C., Davies, M.E.P.: Harmony generation driven by a perceptually motivated tonal interval space. ACM Comput. Entertain. **14**(2), 6 (2016)
4. Bernardes, G., Davies, M., Guedes, C.: Audio key estimation with adaptive mode bias. In: Proceedings of ICASSP, pp. 316–320 (2017)
5. Bidelman, G.M., Krishnan, A.: Brainstem correlates of behavioral and compositional preferences of musical harmony. Neuroreport **22**, 212–216 (2011)
6. Brent, W.: A timbre analysis and classification toolkit for pure data. In: Proceedings of ICMC, pp. 224–229 (2010)
7. Cook, N.: Harmony, Perspective, and Triadic Cognition. Cambridge University Press, Cambridge (2012)
8. Davies, M., Stark, A., Gouyon, F., Goto, M.: Improvasher: a real-time mashup system for live musical input. In: Proceedings of NIME, pp. 541–544 (2014)
9. Davies, M.E.P., Hamel, P., Yoshii, K., Goto, M.: Automashupper: automatic creation of multi-song music mashups. IEEE Trans. ASLP **22**(12), 1726–1737 (2014)
10. Engel, J., et al.: Neural audio synthesis of musical notes with WaveNet autoencoders. In: Proceedings of the 34th International Conference on Machine Learning, pp. 1068–1077 (2017)
11. Euler, L.: Tentamen novae theoriae musicae. Broude (1968/1739)
12. Gebhardt, R., Davies, M., Seeber, B.: Psychoacoustic approaches for harmonic music mixing. Appl. Sci. **6**(5), 123 (2016)
13. Griffin, G., Kim, Y., Turnbull, D.: Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. In: Proceedings of ICASSP, pp. 437–440 (2010)
14. Huron, D.: Interval-class content in equally tempered pitch-class sets: common scales exhibit optimum tonal consonance. Music Percept. **11**(3), 289–305 (1994)
15. Hutchinson, W., Knopoff, L.: The acoustic component of western consonance. J. New Music Res. **7**(1), 1–29 (1978)
16. Johnson-Laird, P.N., Kang, O.E., Leong, Y.C.: On musical dissonance. Music Percept. **30**(1), 19–35 (2012)
17. Krumhansl, C.L., Kessler, E.J.: Tracing the dynamic changes in perceived tonal organisation in a spatial representation of musical keys. Psychol. Rev. **89**, 334–368 (1982)
18. Lahdelma, I., Eerola, T.: Mild dissonance preferred over consonance in single chord perception. i-Perception (2016). https://doi.org/10.1177/2041669516655812
19. Lee, C.L., Lin, Y.T., Yao, Z.R., Lee, F.Y., Wu, J.L.: Automatic mashup creation by considering both vertical and horizontal mashabilities. In: Proceedings of ISMIR, pp. 399–405 (2015)
20. Manovich, L.: The Language of New Media. MIT Press, Cambridge (2001)
21. Mixed in Key: Mashup 2 [software]. http://mashup.mixedinkey.com. Accessed 28 Mar 2017

22. Native Instruments: Traktor pro 2 [software]. https://www.native-instruments.com/en/products/traktor/dj-software/traktor-pro-2/. Accessed on 1 Sep 2017
23. Plazak, J., Huron, D., Williams, B.: Fixed average spectra of orchestral instrument tones. Empirical Musicol. Rev. **5**(1), 10–17 (2010)
24. Roads, C.: Microsound. MIT Press, Cambridge (2004)
25. Schwartz, D.A., Howe, C., Purves, D.: The statistical structure of human speech sounds predicts musical universals. J. Neurosci. **23**(18), 7160–7168 (2003)
26. Sha'ath, I.: Estimation of key in digital music recordings. Master's thesis, Birkbeck College, University of London (2011)
27. Shiga, J.: Copy-and-persist: the logic of mash-up culture. Crit. Stud. Media Commun. **24**(2), 93–114 (2007)
28. Zwicker, E., Fastl, H.: Psychoacoustics-Facts and Models. Springer, Heidelberg (1990). https://doi.org/10.1007/978-3-540-68888-4

TIV.LIB: AN OPEN-SOURCE LIBRARY FOR THE TONAL DESCRIPTION OF MUSICAL AUDIO

António Ramires

Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
antonio.ramires@upf.edu

Gilberto Bernardes

INESC TEC and University of Porto
Faculty of Engineering
Porto, Portugal
gba@fe.up.pt

Matthew E. P. Davies

University of Coimbra
CISUC, DEI
Coimbra, Portugal
mepdavies@dei.uc.pt

Xavier Serra

Music Technology Group
Universitat Pompeu Fabra
Barcelona, Spain
xavier.serra@upf.edu

ABSTRACT

In this paper, we present `TIV.lib`, an open-source library for the content-based tonal description of musical audio signals. Its main novelty relies on the perceptually-inspired Tonal Interval Vector space based on the Discrete Fourier transform, from which multiple instantaneous and global representations, descriptors and metrics are computed—e.g., harmonic change, dissonance, diatonicity, and musical key. The library is cross-platform, implemented in Python and the graphical programming language Pure Data, and can be used in both online and offline scenarios. Of note is its potential for enhanced Music Information Retrieval, where tonal descriptors sit at the core of numerous methods and applications.

1. INTRODUCTION

In Music Information Retrieval (MIR), several libraries for musical content-based audio analysis, such as Essentia [1], Librosa [2], and madmom [3] have been developed. These libraries have been widely adopted across academia and industry as they promote the fast prototyping of experimental methods and applications ranging from large-scale applications such as audio fingerprinting and music recommendation, to task-specific MIR analysis including chord recognition, structural segmentation, and beat tracking.

The tonal domain of content-based audio descriptors denotes all attributes related to the vertical (i.e., harmonic) and horizontal (i.e., melodic and voice-leading) combination of tones, as well as their higher-level governing principles, such as the concept of musical key. The earliest research in this domain was driven by the methods applied to symbolic representations of music, e.g., MIDI files. The jump from symbolic to musical audio domain raises significant problems and requires dedicated methods, as polyphonic audio-to-symbolic transcription remains a challenging task [4]. While the state of the art [5, 6] in polyphonic music transcription has advanced greatly due to the use of deep neural networks, it remains largely restricted to piano-only recordings.

One of the most prominent tonal audio descriptors is the chroma vector. This representation divides the energy of the spectrum of an audio signal in the 12 tones of the western chromatic scale across all octaves. This leads to a 12-element vector where each element corresponds to the energy of each pitch class. Throughout this work, this vector will be referred to as the pitch profile. Many algorithms for this representation have been proposed, including Pitch Class Profiles [7], Harmonic Pitch Class Profiles (HPCP) [8], the CRP chroma [9], and the NNLS chroma [10]. Stemming from this 12-element vector, many metrics and systems have been proposed, for key detection, chord recognition, cover song identification, mood recognition, and harmonic mixing. Yet, despite their fundamental role in many MIR tasks, tonal descriptors are not only less prominent in existing content-based audio libraries, in comparison with rhythmic or timbral descriptors [1], but also their perceptual basis is of limited scope [11].

In the context of the aforementioned limitations, we present the `TIV.lib`, a cross-platform library for Python and Pure Data, which automatically extracts multiple perceptually-aware tonal descriptions from polyphonic audio signals, without requiring any audio-to-symbolic transcription stage. It owes its conceptual basis to ongoing work within music theory on the DFT of pitch profiles, which has been extended to the audio domain [12]. The hierarchical nature of the Tonal Interval Vector (TIV) space allows the computation of instantaneous and global tonal descriptors including harmonic change, (intervallic) dissonance, diatonicity, chromaticity, and key, as well as the use of distance metrics to extrapolate different harmonic qualities across tonal hierarchies. Furthermore, it can enable the efficient retrieval of isolated qualities or those resulting from audio mixes in large annotated datasets as a simple nearest neighbour-search problem.

The remainder of this paper is organized as follows. Section 2 provides an overview of the ongoing work on pitch profiles DFT-based methods within music theory, followed by a description of the recently proposed TIV space, which extends the method to the audio domain. Section 3 provides a global perspective of the newly proposed `TIV.lib` architecture. Section 4 details the mathematical and musical interpretation of the description featured in `TIV.lib` and, finally, Section 5 discusses the scope of application scenarios of the library and Section 6 provides perspectives on future work.

2. RELATED WORK

2.1. Tonal pitch spaces

Within the research literature, numerous tonal pitch spaces and pitch distance metrics have been proposed [13, 14, 15, 16]. They aim to capture perceptual musical phenomena by geometrical and algebraic representations, which quantify and (visually) represent pitch proximity. These spaces process pitch as symbolic manifestations, thus capturing musical phenomena under very controlled conditions, with some of the most prominent spaces discarding the pitch height dimension by collapsing all octaves into a 12-tone pitch space.

Attempts to represent musical audio in the aforementioned spaces have been pursued [17, 18] by adopting an audio-to-symbolic transcription stage. Yet, polyphonic transcription from musical audio remains a challenging task which is prone to error.

Recently, Bernardes et al. [12] proposed a tonal pitch space which maps chroma vectors derived from audio signals driven into a perceptually inspired DFT space. It expands the aforementioned pitch spaces with strategies to process the timbral/spectral information from musical audio.

2.2. From the DFT of symbolic pitch distributions to the Tonal Interval Vector space

In music theory, the work proposed by Quinn [19] and followed by [20, 21, 22] on the Discrete Fourier Transform (DFT) of pitch profiles, has been shown to elicit many properties with music-theoretic value. Moreover, in [23] DFT-based pitch spaces were shown to capture human perceptual principles.

In the Fourier space, a 6-element complex vector, corresponding to the $1 \leq k \leq 6$ DFT coefficients, is typically adopted. The magnitude of the Fourier coefficients has been used to study the shape of pitch profiles, notably concerning the distribution of their interval content. This allows, for example, to quantify diatonic or chromatic structure (see Section 4 for a comprehensive review of the interpretations of the coefficients). The phase of the pitch profiles in the Fourier space reveals aspects of tonal music in terms of voice-leading [24], tonal regions modelling and relations [25], and the study of tuning systems [20]. In summary, the magnitude of the pitch profiles express harmonic quality and the phases harmonic proximity.

Recently, a perceptually-inspired equal-tempered, enharmonic, DFT-based TIV space [12] was proposed. One novelty introduced by this newly proposed space in relation to remaining Fourier spaces was the combined use of the six coefficients in a TIV, $T(k)$. Moreover, the perceptual basis of the space is guaranteed by weighting each coefficients by empirical ratings of dyad consonance, $w_a(k)$. $T(k)$ allows the representation of hierarchical or multi-level pitch due to the imposed L_1 norm, such that:

$$T(k) = w_a(k) \sum_{n=0}^{N-1} \bar{c}(n) e^{\frac{-j2\pi kn}{N}}, \quad k \in \mathbb{Z} \quad \text{with} \quad \bar{c}(n) = \frac{c(n)}{\sum_{n=0}^{N-1} c(n)} \tag{1}$$

where N=12 is the dimension of the chroma vector, $c(n)$, and k is set to $1 \leq k \leq 6$ for $T(k)$ since the remaining coefficients are symmetric. The weights, $w_a(k) = \{3, 8, 11.5, 15, 14.5, 7.5\}$, adjust the contribution of each dimension k of the space to comply with empirical ratings of dyad consonance as summarised in [26]. $w_a(k)$ accounts for the harmonic structure of musical audio driven from an average spectrum of orchestral instruments [11].

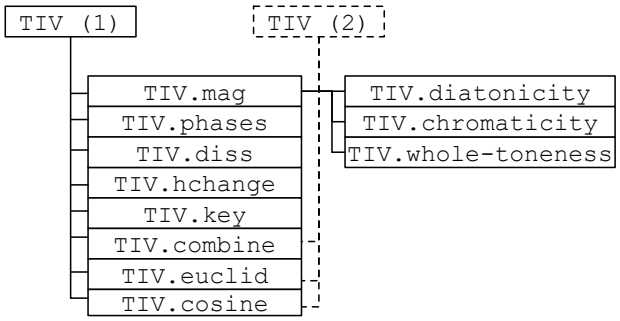


Figure 1: A graph of the dependencies of the feature extraction modules of `TIV.lib`. The algorithms connected to TIV(2) through a dashed line require two inputs for the feature calculation.

3. TIV.LIB: IMPLEMENTATION

The `TIV.lib` includes several signal-processing functions or descriptors for characterising the tonal content of musical audio. This library is implemented in Python, using only `Numpy` and `Scipy` as dependencies and Pure Data, with both available to download at: `http://bit.ly/2pBYhqZ`. Illustrative analysis of musical audio examples for the descriptors are provided in the Python download link as a Jupyter Notebook. The Python implementation targets batch offline processing and the Pure Data implementation online processing.

As an input, the library takes 12-element chroma vectors, $c(n)$, from which TIVs, $T(k)$, are then computed. [1] Any input representation will have an effect on the space, as such we leave the choice of which chroma representation up to the user in order to best fit the problem at hand. Although the system is agnostic to the chosen chroma, we recommend the "cleanest" chroma representation, i.e., that which is closest to a symbolic representation, to be selected. The time scale of the TIV is dependent of the adopted window size during the chroma vector computation. For instantaneous TIVs, a single-window chroma vector can be used as input. For global TIVs, consecutive chroma vectors can be averaged across the time axis prior to the TIV computation.

In Figure 1 we present the architecture of `TIV.lib`. In this graph of dependencies we can see the algorithms that have been implemented and which classes they require for their calculation.

4. TIV.LIB: ALGORITHMS

This section details the functions included in the `TIV.lib`, focusing on their mathematical definition and musical interpretation.

`TIV` is a 6-element complex vector, which transforms chroma into an interval vector space by applying Eq. 1, an L_1-norm weighted DFT. The resulting space dimensions combine intervallic information in the coefficients' magnitude and the tonal region (i.e., musical key area) it occupies in the coefficients' phase. The

[1] A tutorial example on the extraction of HPCP representations from audio is provided in the library package, both using Essentia and Librosa.

mapping between chroma and the TIV retains the bijective property of the DFT and allows the representation of any variable-density pitch profile in the chroma space as a unique location in the TIV space.

`TIV.mag` is a 6-element (real) vector that reports the magnitude of the TIV elements $1 \leq k \leq 6$, such that:

$$mag(k) = ||T(k)|| \tag{2}$$

It provides a characterisation of the harmonic quality of a pitch profile, namely its intervallic content, distilling the same information as the pitch-class interval vector [27, 28]. Mathematically, it is well-understood that a large magnitude in $T(k)$ coefficients indicates how evenly the pitch profile can be divided by N/k. Musically, the work on the DFT of pitch profiles [12, 19] emphasizes the association between the magnitude of Fourier coefficients and tonal qualities: $||T(1)|| \leftrightarrow chromaticity$, $||T(2)|| \leftrightarrow dyadicity$, $||T(3)|| \leftrightarrow triadicity$, $||T(4)|| \leftrightarrow diminished\,quality$, $||T(5)|| \leftrightarrow diatonicity$, $||T(6)|| \leftrightarrow whole-toneness$. Please refer to [20, 21] for a comprehensive discussion on the interpretation of the DFT coefficients.[2] One distinct property of the `TIV.mag` vector is its invariance under transposition or inversion [20]. For example, all major triads or harmonic minor scales share the same Fourier magnitude, hence the same `TIV.mag` vector [3].

`TIV.phases` is a 6-element (real) vector that reports the phases (or direction) of the TIV coefficients $1 < k < 6$, such that:

$$phases(k) = \angle T(k) \tag{3}$$

It indicates which of the transpositions of a pitch profile quality is under analysis [29], as transposition of a pitch profile by p semitones, i.e., circular rotations of the chroma, $c(n)$, rotates the $T(k)$ by $\varphi(p) = \frac{-2\pi kp}{N}$. TIV phases are also associated with regional (or key) areas, whose diatonic set is organised as clusters in the TIV space [12, 25].

`TIV.combine` computes the resulting TIV from mixing (or summing) multiple TIVs representing different musical audio signals. Due to the properties of the DFT space, this operation can be efficiently computed as a linear combination of any number of TIVs, $T(k)$. Given TIVs $T_1(k)$ and $T_2(k)$, their linear combination, weighted by their respective energy, a_1 and a_2, is given by:

$$T_{1+2}(k) = \frac{T_1(k) \cdot a_1 + T_2(k) \cdot a_2}{a_1 + a_2} \tag{4}$$

a_1 and a_2 are retrieved from the discarded DC components $T_1(0)$ and $T_2(0)$.

`TIV.chromaticity` reports the level of concentration of a sonority in a specific location of the chromatic pitch circle as a value within the [0,1] range, computed as the magnitude of the $T(1)$ normalized to unity: $\frac{||T(1)||}{w_a(1)}$. This value is close to 0 for sounds exhibiting energy in evenly-spaced pitch classes (such as typically tonal chords and scales) and close to 1 for chromatic pitch aggregates.

`TIV.diatonicity` reports the level of concentration of a sonority within the circle of fifths as a value within the [0,1] range.

[2] We note for each of these single Fourier coefficient quantities that the effects of the weights can be factored out.

[3] Note that the phases, discarded here, will differ. As such, the uniqueness property of the TIV is maintained as it combines both magnitude and phase information.

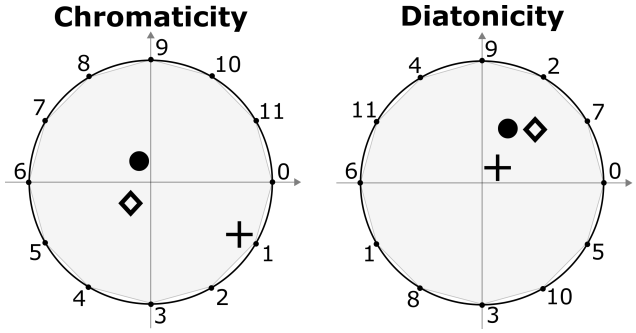


Figure 2: Two DFT coefficients interpreted as chromaticity and diatonicy. Three TIV are plotted for comparison: C major chord {0,4,7} (◇), 3-note chromatic cluster {0,1,2} (+), and C major scale {0,2,4,5,7,9,11} (●). The grey shaded areas indicate the space the TIVs can occupy.

The larger the magnitude of the $T(5)$ normalized to unity, $\frac{||T(5)||}{w_a(5)}$, the higher the level of diatonicity.

`TIV.whole-toneness` reports the proximity to one of the two existing whole-tone collection within the 12-tone equal temperament tuning. The level of whole-toneness is reported within the [0,1] range resulting from the magnitude of the $T(6)$ normalized to unity, such that: $\frac{||T(6)||}{w_a(6)}$.

Fig. 2 shows the DFT coefficients from which we extract chromaticity and diatonicity descriptions as the magnitude of $T(1)$ and $T(5)$, respectively. We plot pitch profiles that aim to illustrate the behaviour of each coefficient in eliciting the chromatic and diatonic character of the C major chord and C major scale as well as chromatic 3-tone cluster by inspecting their magnitude. Note that the magnitude of both the C major chord and C major scale, two prototypical diatonic pitch profiles, clearly have greater magnitude in the diatonic $T(5)$ coefficient in comparison with the three-note cluster, a prototypical chromatic profile. Conversely, in the chromatic $T(1)$ coefficient, the magnitudes of the above pitch profiles show the expected opposite behaviour, thus mapping the three-note cluster further from the centre.

`TIV.euclid` and **`TIV.cosine`** compute the Euclidean, E, and cosine, C, distance between two given TIVs, $T_1(k)$ and $T_2(k)$, using Eqs. 5 and 6, respectively.

$$E\{T_1, T_2\} = \sqrt{||T_1 - T_2||^2} \tag{5}$$

$$C\{T_1, T_2\} = \frac{T_1 \cdot T_2}{||T_1||||T_2||} \tag{6}$$

The cosine distance (i.e., the angular distance) between TIVs can be used as an indicator of how well pitch profiles "fit" or mix together. For example, it quantifies the degree of tonal proximity of TIV mixtures, or informs which translation or transposition of a TIV best aligns with a given key. Conversely, Euclidean distances between TIVs relate mostly to melodic (or horizontal) distance. It captures the neighbouring relations observed in the *Tonnetz*, where smaller distances agree with parsimonious movements between pitch profiles. Please refer to [22, 24] for a comprehensive discussion on this topic.

`TIV.hchange` computes a harmonic change detection function across the temporal dimension of an audio signal. Peaks in this function indicate transitions between regions that are harmonically

stable. We compute a harmonic change measure, λ, for an audio frame m as the Euclidean distance between frames $m+1$ and $m-1$ (Eq. 7), an approach inspired by Harte et al. [30], which can be understood as adopting three coefficients out of the $1 \leq k \leq 6$ of the TIV, $T(k)$, i.e., those corresponding to the circle of fifths, the circle of minor thirds, and the circle of major thirds.

$$\lambda_m = \sqrt{||T_{m-1} - T_{m+1}||^2} \tag{7}$$

`TIV.diss` provides an indicator of (interval content) dissonance, as the normalized TIV magnitude subtracted from unity, $1 - \frac{|T(k)|}{|w_a(k)|}$. This perceptually-inspired indicator stems from the weighted magnitude of the TIV coefficients, which rank the intervals $1 \leq k \leq 6$ to match empirical ratings of dissonance within the Western tonal music context [11, 12].

`TIV.key` infers the key from an audio signal as a pitch class (tonic) and a mode (major or minor). It is computed as the Euclidean distance from the 24 major and minor key TIVs, $T_r^{p\star}(k)$, defined as the shifts (i.e. rotation) of the 12 major and 12 minor profiles, p, by Temperley [31] or Sha'ath [32], such that:

$$R_{min} = \text{argmin}_r \sqrt{||T \cdot \alpha - T_r^{p\star}||^2} \tag{8}$$

where $T_r^{p\star}$ are 24 major and minor key profiles TIVs, p. When $r \leq 11$, we adopt the major profile and when $r \geq 12$, the minor profile. α is a bias introduced to balance the distance between major and minor keys. Optimal values of $\alpha = 0.2$ and $\alpha = 0.55$ have been proposed in [33] for the Temperley [31] and Shat'ath [32] key profiles, respectively. The output is an integer, R_{min}, ranging between $0-11$ for major keys and $12-23$ for minor keys, where 0 corresponds to C major, 1 to C# major, and so on through to 23 being B minor.

5. APPLICATIONS AND PERSPECTIVES

Following the emerging body of music theory literature on the DFT of pitch profiles and the continuous work of the TIV space, we implemented the perceptually-inspired `TIV.lib` in Python and Pure Data. The former aims at batch offline processing and the latter mostly at online or real-time processing, but allowing offline computations as well.

Figure 3 shows an example usage of the `TIV.lib` functions for computing the diatonicity, chromaticity and whole-toneness harmonic qualities in Pure Data.

In order to achieve the same result in Python, the following code can be executed:

```
import TIVlib as tiv

ex_tiv = tiv.TIV.from_pcp(example_chroma)
ex_wholetoneness = ex_tiv.wholetoneness()
ex_diatonicity = ex_tiv.diatonicity()
ex_chromacity = ex_tiv.chromaticity()
```

We now provide an example usage of this library for extracting tonal features of a musical piece. We run this code for an excerpt of the Kraftwerk song "Spacelab," to demonstrate how this library can provide useful information related to its diatonicity and whole-toneness. As can be seen from the Chromagram in Figure 4, this song starts in the whole tone scale [F# G# A#C D E], and then moves, at 33 s, to a diatonic set [C D Eb

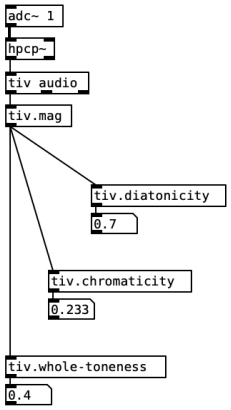


Figure 3: Pure Data patch for online computation of diatonicity, chromaticity and whole-toneness harmonic qualities of a live input audio signal, using the `TIV.lib`.

F G Ab Bb C]. In Figure 5 we show this by plotting the evolution of the`TIV.diatonicity`, `TIV.wholetoneness` and `TIV.chromaticity` outputs for this music.

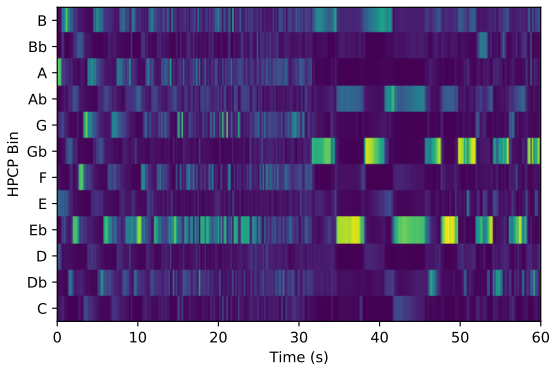


Figure 4: Chromagram of the first minute of Kraftwerk's "Spacelab."

Several functions of the `TIV.lib` result from ongoing research and have been evaluated in previous literature [11, 12, 33, 34], where the possibility of the TIV space to geometrically and algebraically capture existing spaces of perceptual and music theoretical value, such as Euler and Krumhansl [12], were shown. In particular, we highlight our previous work on key recognition [33] and harmonic mixing [11, 35], where TIV-based approaches outperformed more traditionally used harmonic features. In addition, the use of TIVs can also extend content-based audio processing by providing a vector space where distances and metrics (e.g., dissonance and harmonic proximity) among multi-level pitch, chords, and keys, capture perceptual aspects of musical phenomena. Examples of creative possibilities of the TIV space have also been shown in Musikverb [36], where it was used for developing a novel

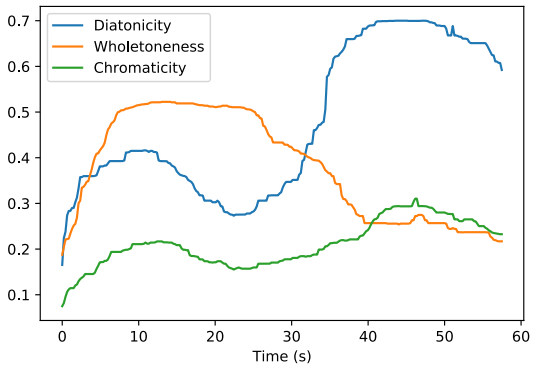


Figure 5: Output of the diatonicity, chromaticity and wholetoneness harmonic qualities the first minute of Kraftwerk's "Spacelab", using the `TIV.lib`.

type of harmonically-adaptive reverb effect.

We strongly believe that the properties of the TIV space can be further explored in content-based audio processing. For example, the possibility to isolate the harmonic quality in `TIV.mag` as a pitch-invariant audio representation can be relevant for several MIR tasks that rely on multiple transposed versions of a given musical pattern, such as in query-by-humming, and cover song detection. Moreover, the possibility to compute TIV mixes as a computationally efficient linear combination allows for the fast retrieval of musical audio from large datasets (e.g., Freesound [37]), as a simple nearest-neighbour search problem. Finally, the newly proposed indicators of tonal quality such as `TIV.chromaticity`, `TIV.diatonicity`, `TIV.wholte-toneness`, and `TIV.diss` not only extend musical theoretical methodologies to content-based processing from audio performance data, but can also promote a greater understanding of tonal content in MIR tasks.

By providing streamlined access to a set of music theoretic properties which are non-trivial to obtain from commonly used time-frequency representations in MIR such as the STFT (or even from chroma-like representations directly), we believe the `TIV.lib` can lay the foundation for a kind of "enhanced" MIR in tasks such as chord recognition and key estimation which can directly leverage the complementary contextual information contained within the `TIV.lib` descriptors.

6. CONCLUSIONS

In this paper we have introduced the open-source tool, `TIV.lib`, as a means to drive the uptake and usage of the Tonal Interval Space both in offline music signal analysis via the python implementation, as well as in online contexts using Pure Data. While we hope to see a growth of applications which benefit from access to music theoretic harmonic features provided by `TIV.lib` our own future work will focus in two principal areas: i) investigating the processing stages which directly precede the calculation of the TIV; and ii) in the application of the TIV across large datasets. More specifically, we seek to study the impact of different methods for calculating the requisite chroma vectors (e.g., HPCP [8], NNLS [10], or timbre-invariant chroma [9]) in the TIV space, as pursued in [33, 34] within the scope of audio key detection. Furthermore, we will study an optimal strategy to define the weights, $w_a(k)$, for particular audio sources and to implement the descriptors in a large online musical database supported by content-based analysis, as a strategy to study the descriptors under a large-scale environment for musical retrieval and creation. Finally, we intend to add this library to existing musical audio analysis libraries such as Essentia and Librosa.

7. ACKNOWLEDGMENTS

António Ramires is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No765068, MIP-Frontiers.

Gilberto Bernardes is supported by Experimentation in music in Portuguese culture: History, contexts, and practices in the 20th and 21st centuries—Project co-funded by the European Union, through the Operational Program Competitiveness and Internationalization, in its ERDF component, and by national funds, through the Portuguese Foundation for Science and Technology.

This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020, as well as by Portuguese National Funds through the FCT - Foundation for Science and Technology, I.P., under the project IF/01566/2015.

8. REFERENCES

[1] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra, "Essentia: an open-source library for sound and music analysis," in *Proc. of the ACM International Conference on Multimedia*, 2013, pp. 855–858.

[2] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in python," in *Proceedings of the Python in Science Conference*, 2015, pp. 109–114.

[3] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new Python Audio and Music Signal Processing Library," in *Proceedings of the 24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, 10 2016, pp. 1174–1178.

[4] E. Benetos, S. Dixon, Z. Duan, and S. Ewert, "Automatic music transcription: An overview," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 20–30, 2018.

[5] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proc. of the 19th Intl. Society for Music Information Retrieval Conf. (ISMIR)*, 2018, pp. 50–57.

[6] A. Ycart and E. Benetos, "Polyphonic music sequence transduction with meter-constrained lstm networks," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 386–390.

[7] T. Fujishima, "Realtime chord recognition of musical sound: a system using common lisp music," in *Proc. of the International Computer Music Conference*, 1999.

[8] E. Gómez, "Tonal description of polyphonic audio for music content processing," *INFORMS Journal on Computing*, vol. 18, no. 3, pp. 294–304, 2006.

[9] M. Müller and S. Ewert, "Towards timbre-invariant audio features for harmony-based music," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 3, pp. 649–662, 2010.

[10] M. Mauch and S. Dixon, "Approximate note transcription for the improved identification of difficult chords," in *Proc. of the International Society for Music Information Retrieval Conference*, 2010, pp. 135–140.

[11] G. Bernardes, M. E. P. Davies, and C. Guedes, "A hierarchical harmonic mixing method," in *Music Technology with Swing*, M. Aramaki, M. E. P. Davies, R. Kronland-Martinet, and S. Ystad, Eds. 2018, pp. 151–170, Springer International Publishing.

[12] G. Bernardes, D. Cocharro, M. Caetano, C. Guedes, and M. E. P. Davies, "A multi-level tonal interval space for modelling pitch relatedness and musical consonance," *Journal of New Music Research*, vol. 45, no. 4, pp. 281–294, 2016.

[13] R. Shepard, "The analysis of proximities: multidimensional scaling with an unknown distance function. i.," *Psychometrika*, vol. 27, no. 2, pp. 125–140, 1962.

[14] F. Lerdahl, *Tonal pitch space*, Oxford University Press, 2004.

[15] D. Tymoczko, *A geometry of music: Harmony and counterpoint in the extended common practice*, Oxford University Press, 2010.

[16] E. Chew, "Out of the grid and into the spiral: Geometric interpretations of and comparisons with the spiral-array model," *Computing in musicology*, vol. 15, pp. 51–72, 2007.

[17] C. Chuan and E. Chew, "Polyphonic audio key finding using the spiral array ceg algorithm," in *IEEE International Conference on Multimedia and Expo*, 2005, pp. 21–24.

[18] B. De Haas, R. Veltkamp, and F. Wiering, "Tonal pitch step distance: a similarity measure for chord progressions.," in *Proc. of the International Society for Music Information Retrieval Conference*, 2008, pp. 51–56.

[19] I. Quinn, "General equal-tempered harmony: parts 2 and 3," *Perspectives of New Music*, pp. 4–63, 2007.

[20] E. Amiot, *Music through Fourier space*, Springer, 2016.

[21] J. Yust, "Stylistic information in pitch-class distributions," *Journal of New Music Research*, vol. 48, no. 3, pp. 217–231, 2019.

[22] D. Tymoczko and J. Yust, "Fourier phase and pitch-class sum," in *International Conference on Mathematics and Computation in Music*, 2019, pp. 46–58.

[23] M. R. W. Dawson, A. Perez, and S. Sylvestre, "Artificial neural networks solve musical problems with Fourier phase spaces," *Scientific Reports*, vol. 10, no. 1, pp. 7151, Apr 2020.

[24] D. Tymoczko, "Set-class similarity, voice leading, and the fourier transform," *Journal of Music Theory*, vol. 52, no. 2, pp. 251–272, 2008.

[25] J. Yust, "Probing questions about keys: Tonal distributions through the DFT," in *International Conference on Mathematics and Computation in Music*, 2017, pp. 167–179.

[26] David Huron, "Interval-class content in equally tempered pitch-class sets: Common scales exhibit optimum tonal consonance," *Music Perception*, vol. 11, no. 3, pp. 289–305, 1994.

[27] A. Forte, "A theory of set-complexes for music," *Journal of Music Theory*, vol. 8, no. 2, pp. 136–183, 1964.

[28] A. Forte, *The structure of atonal music*, vol. 304, Yale University Press, 1973.

[29] J. Hoffman, "On pitch-class set cartography: Relations between voice-leading spaces and fourier spaces," *Journal of Music Theory*, vol. 52, no. 2, pp. 219–249, 2008.

[30] C. Harte, M. Sandler, and M. Gasser, "Detecting harmonic change in musical audio," in *Proc. of the ACM Workshop on Audio and Music Computing Multimedia*, 2006, pp. 21–26.

[31] D. Temperley, "What's key for key? the Krumhansl-Schmuckler key-finding algorithm reconsidered," *Music Perception: An Interdisciplinary Journal*, vol. 17, no. 1, pp. 65–100, 1999.

[32] I. Sha'ath, "Estimation of key in digital music recordings," M.S. thesis, Birkbeck College, University of London, 2011.

[33] G. Bernardes, M. E. P. Davies, and C. Guedes, "Automatic musical key estimation with adaptive mode bias," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 316–320.

[34] G. Bernardes, D. Cocharro, C. Guedes, and M. E. P. Davies, "Harmony generation driven by a perceptually motivated tonal interval space," *ACM Computers in Entertainment*, vol. 14, no. 2, pp. 6, 2016.

[35] C. Maçãs, A. Rodrigues, G. Bernardes, and P. Machado, "Mixmash: An assistive tool for music mashup creation from large music collections," *International Journal of Art, Culture and Design Technologies*, vol. 8, no. 2, pp. 20–40, 2019.

[36] J. Pereira, G. Bernardes, and R. Penha, "Musikverb: A harmonically adaptive audio reverberation," in *Proc. of the 21st Int. Conference on Digital Audio Effects (DAFx-18), Aveiro, Portugal*, Sep 2018.

[37] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. of the ACM International Conference on Multimedia*, 2013, pp. 411–412.

[38] G. Bernardes, M. E. P. Davies, and C. Guedes, "Automatic musical key estimation with adaptive mode bias," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 316–320.

[39] R. Gebhardt, M. E. P. Davies, and B. Seeber, "Harmonic mixing based on roughness and pitch commonality," in *Proc. of the 18th Int. Conference on Digital Audio Effects (DAFx-15), Trondheim, Norway*, Nov 2015, pp. 185–192.

On Filter Generalization for Music Bandwidth Extension Using Deep Neural Networks

Serkan Sulun and Matthew E. P. Davies

***Abstract*—In this paper, we address a subtopic of the broad domain of audio enhancement, namely musical audio bandwidth extension. We formulate the bandwidth extension problem using deep neural networks, where a band-limited signal is provided as input to the network, with the goal of reconstructing a full-bandwidth output. Our main contribution centers on the impact of the choice of low-pass filter when training and subsequently testing the network. For two different state-of-the-art deep architectures, ResNet and U-Net, we demonstrate that when the training and testing filters are matched, improvements in signal-to-noise ratio (SNR) of up to 7 dB can be obtained. However, when these filters differ, the improvement falls considerably and under some training conditions results in a lower SNR than the band-limited input. To circumvent this apparent overfitting to filter shape, we propose a data augmentation strategy which utilizes multiple low-pass filters during training and leads to improved generalization to unseen filtering conditions at test time.**

***Index Terms*—Audio bandwidth extension, audio enhancement, deep neural networks, generalization, regularization, overfitting.**

I. Introduction

MODERN recording techniques provide music signals with extremely high audio quality. By contrast, the listening experience of archive recordings, such as jazz, pop, folk, and blues recorded before the 1960s is arguably limited by the recording techniques of the time as well as the degradation of physical media. Even so, modern recordings can also suffer from diminished audio quality due to the use of lossy compression, downsampling, packet loss, or clipping. In the broadest sense, audio enhancement aims to restore a degraded signal to improve its sound quality [1]. As such, audio enhancement may target the removal of noise, the suppression of cracks or pops (e.g. from old vinyl records), signal completion to fill in gaps (so-called "audio inpainting" [2], [3]), or the bandwidth extension of a band-limited signal.

To transmit audio signals through internet streams, or for the ease of storing, common operations such as compression, bandwidth reduction, and low-pass filtering all result in the removal of at least part of the high-frequency audio content. Optionally, the signal can be downsampled afterwards, effectively reducing its size. While this process can be understood as a relatively straightforward mapping from a *full-bandwidth*, or *wideband* signal to a *band-limited* or *narrowband* signal, the corresponding inverse problem, namely *bandwidth extension*, seeks to reconstruct missing high-frequency content and is thus non-trivial. Furthermore, if the input signal is downsampled, the inverse problem also requires upsampling, and the overall process is called *super-resolution*, a term that is commonly used in the image processing literature. Despite these challenges, bandwidth extension is crucial for increasing the fidelity of audio, especially for speech and music signals.

The first applications of audio bandwidth extension addressed speech signals only, due to the practical problems arising from the low bandwidth of telephone systems. One of the earliest works used a statistical approach in which narrowband and wideband spectral envelopes were assumed to be generated by a mixture of narrowband and wideband sources [4]. Codebook mapping-based methods use two learned codebooks, belonging to the narrowband and wideband signals, containing spectral envelope features, where a one-to-one mapping exists between their entries [5], [6]. In linear mapping-based methods, a transformation matrix is learned using methods such as least-squares [7], [8].

Later methods sought to learn to model the wideband signal directly, rather than the mapping between predefined features. Gaussian mixture models (GMMs) have been used to estimate the joint probability density of narrowband and wideband signals [9], [10]. Other approaches include the use of hidden Markov models (HMMs), where each state of the model represents the wideband extension of its narrowband input [11]–[13]. Due to their recursive mechanism, HMMs can leverage information from the past input frames. Methods based on non-negative matrix factorization (NMF) model the speech signals as a combination of learned non-negative bases [14], [15]. In the testing stage, low-frequency base components of the input can be used to estimate how to combine the high-frequency base components to create the wideband signal. Finally, the first works using neural

Manuscript received May 1, 2020; revised September 14, 2020; accepted October 30, 2020. Date of publication November 16, 2020; date of current version January 29, 2021. The work of Serkan Sulun was supported by la Caixa Foundation (ID 100010434), under Fellowship Code LCF/BQ/DI19/11730032. This work was supported in part by national funds through the FCT - Foundation for Science and Technology, I.P., under Project CISUC - UID/CEC/00326/2020, in part by European Social Fund, through the Regional Operational Program Centro 2020, and in part by Portuguese National Funds through the FCT - Foundation for Science and Technology, I.P., under Project IF/01566/2015. The guest editor coordinating the review of this manuscript and approving it for publication was Dr. N. Holighaus. *(Corresponding author: Serkan Sulun.)*

Serkan Sulun is with the Institute for Systems and Computer Engineering, Technology and Science (INESC TEC), 4200-465 Porto, Portugal (e-mail: serkan.sulun@inesctec.pt).

Matthew E. P. Davies is with the University of Coimbra, Centre for Informatics and Systems of the University of Coimbra, Department of Informatics Engineering, 3030-790 Coimbra, Portugal (e-mail: mepdavies@dei.uc.pt).

Digital Object Identifier 10.1109/JSTSP.2020.3037485

networks for speech bandwidth extension employed multilayer perceptrons (MLPs) to estimate linear predictive coding (LPC) coefficients of the wideband speech signal [16], or to find a shaping function to transform the spectral magnitude [17]. We note that these early works used very small neural networks, in which the total number of parameters was around 100.

More recent approaches to audio bandwidth extension have used deep neural networks (DNNs), with many more layers and far greater representation power than their older counterparts. DNNs also eliminate the need for hand-crafted features, as they can use raw audio or time-frequency transforms as input, and then learn appropriate intermediate representations. Early works using DNNs on speech bandwidth extension employed audio features as inputs, and demonstrated the superiority of DNNs over the state-of-the-art method of the time, namely GMMs [18], [19]. Another pioneering DNN-based work used the frequency spectrogram as the input [20]. A much deeper model employed the popular *U-Net* architecture [21] and works in the raw audio domain, performing experiments on both speech and single instrument music [22]. Lim *et al.* combined the two aforementioned approaches creating a dual network, which operates separately in the time and frequency domains, and creates the final output using a fusion layer [23]. A recent work used the *U-Net* in the time domain only, but the training loss was a combination of losses calculated in both the time and frequency domains [24]. To increase the qualitative performance, namely, the clarity of the produced audio, generative adversarial networks [25] have also been employed in DNN-based audio bandwidth enhancement [26], [27]. The latest work by Google on music enhancement presents an ablation study, using SNR to measure distortion, and VGG distance, namely the distance between the embeddings of the VGGish network [28], as the perceptual score [29]. Their results show that the incorporation of adversarial loss yields a better perceptual score at the expense of decreasing SNR.

II. Motivation and Paper Outline

A. Motivation

While the enhancement of old music recordings can be partially framed in the context of bandwidth extension, certain risks arise when considering the data that DNNs are given for training. Even though trained DNNs can perform well on samples from the training data, they may not exhibit the same performance on unseen samples from the testing data. This phenomenon is named *sample overfitting* and even though it is an important concern, especially for classification tasks, its existence in generative tasks, such as image super-resolution, audio bandwidth extension, and adversarial generation, is debated. Recent studies show that sample overfitting is not observed for both discriminators and generators of generative adversarial networks [30], [31], and supervised generative networks for video frame generation [32]. Furthermore, state-of-the-art image super-resolution networks do not include any regularization layers [33]–[35], such as batch normalization [36] and dropout [37], to avoid overfitting.

Especially in the task of automatic speech recognition, models may not generalize well to audio samples recorded in a completely different environment, even when the speakers remain the same. Methods to resolve this problem are referred in the literature as *multi-environment* [38], *multi-domain* [39], or *multi-condition* [40] approaches, and consist of using training samples recorded in multiple environments, with the goal of generalization to unseen environments. Some works simulate the multiple environment conditions through data pre-processing. One study created training samples by adding noise with different signal-to-noise (SNR) levels on clean speech signals [41]. Another work on speech bandwidth enhancement included input training samples that are created using low-pass filters with different cut-off frequencies [42]. In all aforementioned examples, the samples that illustrate multiple conditions are perceptually different.

Another risk concerns the pre-processing methods used to create the training data. When considering music bandwidth extension for enhancing archive recordings, no full-bandwidth version exists and as such, there is no "ground-truth" target for DNNs. To this end, training data is typically obtained by low-pass filtering full-bandwidth recordings. However, since real-world band-limited samples are not the result of some hypothetical universal digital low-pass filter, it can be challenging to develop robust techniques for bandwidth extension which rely on a loose approximation of the bandwidth reduction process, and in turn to generalize to unseen recordings. While trained DNNs perform well on training data created with one type of low-pass filter, they may fail to generalize to audio content subjected to different types of low-pass filters. This phenomenon can occur even when these different types of low-pass filters have the same cut-off frequency, creating samples that may have almost no perceptual difference. Throughout this paper, we call this *filter overfitting*, which can be understood as a lack of *filter generalization*.

While Kuleshov *et al.* [22] do not explicitly target filter generalization, they present a rudimentary analysis of generalization related to the presence or absence of a pre-processing filter. Their main goal is audio super-resolution, and while preparing their band-limited input data, before downsampling, they optionally use a low-pass filter. They demonstrate results in which a low-pass filter is not present while preparing the input training data, but is present for the test data, and vice-versa. Both training and testing data are still downsampled, hence they investigate the generalization in the context of aliased and non-aliased data. When the aliasing conditions match, the model performs well, with test SNR levels around 30 dB. But when these conditions do not match, the model becomes ineffective, with test SNR levels around 0.4 dB, showing no generalization to the addition or the removal of the low-pass filter during testing.

B. Contributions

While the use of low pass filtering is widespread among existing work on audio bandwidth extension using DNNs, to the best of our knowledge, no work to date has thoroughly investigated the topic of filter generalization. We argue that the lack of generalization to various types of signal deterioration is an important challenge in creating audio enhancement models for real-world deployment. In this work, we present a rigorous

analysis of filter generalization, evaluating generalization to different filters used to pre-process input data, on the task of bandwidth enhancement of complex music signals, using two popular DNN architectures.

To evaluate sample overfitting, we use disjoint testing and training data, to create totally *unseen data* for the trained models. To evaluate filter generalization, we pre-process the testing input data with a filter that does not match the filters that pre-process the training input data, i.e., an *unseen filter* and compare it to the test setting where the filters used for training and testing data do match, i.e, *seen filters*. We argue that testing with the unseen filter can be considered a kind of real-world signal degradation, in which the true underlying degradation function is unknown.

We evaluate three different regularization methods that are used in the literature to increase generalization. In particular, we compare the usage of data augmentation, batch normalization, and dropout, against the baseline of not using any regularization methods. We introduce a novel data augmentation technique of using a set of different low pass filters to pre-process the input data, in which the unseen test filter is never present. We examine the training process by tracking the model's performance throughout training iterations, by performing validation using both seen and unseen filters.

Similar to image super-resolution methods, we use fully-convolutional DNNs to model the raw signal directly [34], [43]. One of the DNN models we employ is the *U-Net*, which was first used for biomedical image segmentation [21], and later in audio signal processing tasks such as singing voice separation [44], and eventually for audio enhancement [22], [26], [45], [46]. In addition to the U-Net, we also use the deep residual network model (*ResNet*) [47] since it is one of the most widely used DNN architectures in signal processing tasks. Even though the U-Net is a popular architecture in the recent audio processing literature, to the best of our knowledge, no work in the domain of audio processing compares the U-Net against the well-established baseline of the ResNet. A small number of comparative studies exist in the fields of image processing and medical imaging, in which either the number of parameters of the compared models is not stated [48], or in which the ResNet has significantly fewer parameters than the U-Net [49]–[51]. In all these works, the ResNet outperforms the U-Net by a small margin. In this paper, we also present a comparison between the U-Net and ResNet, where each network has a similar number of parameters.

Our main findings indicate that filter overfitting occurs for both the U-Net and ResNet, although to different degrees, and that the use of multi-filter data augmentation, as opposed to more traditional regularization techniques, is a promising means to mitigate this overfitting problem and thus improve filter generalization for bandwidth extension.

C. Outline

The remainder of the paper is organized as follows. Section III-A presents the architectures of the baseline models used. Section III-B defines the existing regularization layers for DNNs and introduces our novel data augmentation method. The rest of Section III describes the dataset, evaluation methods, and implementation details. In Section IV we present a detailed analysis of the performance of the trained models. Finally, in Section V we present conclusions and highlight promising areas for future work.

III. Methodology

A. Models

In this section, we define the two baseline models: U-Net and ResNet. For both models, we follow the approach of Kuleshov *et al.* [22] and use raw audio as the input rather than time-frequency transforms (e.g., as in [52]). As such we remove any need for phase reconstruction in the output. However, since we address bandwidth extension and not audio super-resolution, our inputs are not subsampled. Hence the sizes of the input and the output are equal for all our models.

1) U-Net: The U-Net architecture [21], like the auto-encoder, consists of two main groups. The first group contains downsampling layers and is followed in the second group by upsampling layers, as shown in Fig. 1a. In the U-Net, individual downsampling and upsampling layers at the same scale are connected through stacking connections, e.g., the output of the first downsampling convolutional block is stacked with the input of the last upsampling convolutional block.

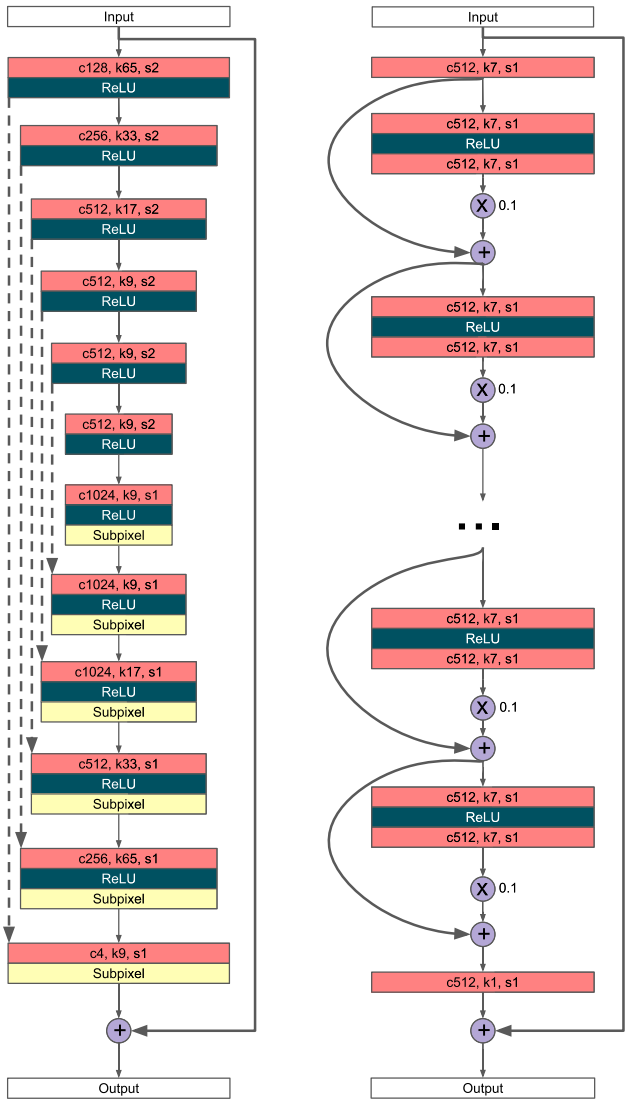
In the downsampling group, one-dimensional convolutional layers with stride 2 are used, effectively halving the activation length. Borrowing from image processing terminology, the upsampling group includes "sub-pixel" layers (also known as the pixel shuffler) [53] to double the activation length. Sub-pixel layers weave the samples in the spatial dimension, taken from alternate channels, effectively halving the channel length.

The number of parameters is selected to replicate the original work using U-Net for audio super-resolution [22], [54], which we denote as *Audio-SR-U-Net* throughout this paper. This resulted in a network with 56.4 million parameters.

2) ResNet: A common issue with training vanilla feed-forward neural networks with many layers is the "vanishing gradient" problem, in which the gradient back-propagated to the earliest layers approaches zero, due to repeated multiplications. Residual networks [47] eliminated this problem by using *residual blocks*, which only model a fraction of the difference between their inputs and outputs. Commonly, each residual block includes two convolutional layers and a nonlinear function in between them. Very deep models include *residual scaling* in which the output of each residual block is multiplied by a small number, e.g., 0.1, and then summed with its input, to further stabilize training. Our ResNet model is represented in Fig. 1b.

Unlike the U-Net, the ResNet activation lengths stay constant throughout the network. In this way, we can avoid any loss of temporal information since our goal is to create a high-resolution output of equal length to the input. Note that we use a simple design where all convolutional layers except the last one have the same number of parameters. Similar in size to the U-Net implementation, it has 55.1 million parameters.

In all our models, all convolutions apply appropriate zero padding to keep the activation sizes constant. This is even true for the downsampling convolutions since the downsampling effect



(a) U-Net model. Dashed lines indicate stacking connections.

(b) ResNet model with 15 residual blocks.

Fig. 1. Models used. c, k, and s indicate channel size, kernel size, and stride of the convolutional layers, respectively.

is achieved using strided convolutions. The *Rectified Linear Unit (ReLU)* is used as the activation function. The loss function for all our models is the mean-squared error. As is common in enhancement models, an additive connection from the input to the output is also used, so that the network only needs to model the *difference* between the input and the target signals, rather than creating the target signal from scratch.

To analyze generalization, we present ablation studies, in which we incorporate common methods to avoid overfitting, defined as *regularization methods*.

B. Regularization Methods

1) Dropout: One of the simplest methods to prevent overfitting is dropout, where activation units are dropped based on a fixed probability [37]. This introduces noise in the hidden layers and prevents excessive co-adaptation.

Although dropout has been largely superseded by batch normalization, especially in residual networks, new state-of-the-art residual models, namely wide residual networks [55] do employ it. Furthermore, *Audio-SR-U-Net*'s open-source implementation [54] uses a dropout layer instead of batch normalization, and thus, we followed this approach in our U-Net model and used dropout layers after each upsampling convolutional layer. In our ResNet model, we placed dropout layers between the two convolutional layers of each residual block. For all experiments, the dropout rate is set to 0.5.

2) Batch Normalization: While training DNNs, updating the parameters of the model effectively changes the distribution of the inputs for the next layers. This is defined as *internal covariate shift* and batch normalization addresses this problem by normalizing the layer inputs [36]. Even though batch normalization is mainly proposed to speed up training, it provides regularization as well. Because the parameters for the normalization are learned based on each batch, they can only provide a noisy estimate of the true mean and variance. Normalization using these estimated parameters introduces noise within the hidden layers and reduces overfitting.

For the U-Net, we follow the *Audio-SR-U-Net* model [22] and insert batch normalization layers after each downsampling convolutional layer. For the ResNet, batch normalization is used after each convolutional layer.

3) Data Augmentation: To increase sample generalization of DNNs, data augmentation is used, where the input data samples are transformed before being fed into the DNN, effectively increasing the number and diversity of training samples. Data augmentation is very common in image-based tasks and mostly utilizes geometric transformations such as rotating, flipping, or cropping [56]. Geometric transformations of this kind when applied to music signals typically do not produce realistic samples. While some work has been conducted on data augmentation for musical signals [57], it primarily targets robustness for classification tasks such as instrument identification in the presence of time-stretching, pitch-shifting, dynamic range compression, and additive noise. Operations of this kind (including minor changes in time or pitch) certainly form part of a larger set of signal degradations that could be explored for musical audio enhancement, however, our focus in this work centers on bandwidth extension and is thus restricted to the consideration of low pass filtering.

Since our main goal in this work is to explore and then improve filter generalization, we propose a data augmentation method where many different types of filters are used during training. Our baseline approach, without data augmentation, uses a *single-filter* training setting, specifically a 6th order Chebyshev Type 1, denoted as "Chebyshev-1, 6". When using data augmentation, in a *multi-filter* training setting, we adopt a set of eight different filters, picked randomly for each input sample during training. These eight filters consist of *Chebyshev-1*, *Bessel*, and *Elliptic* filters of different orders. To evaluate filter generalization, we reserve the 6th order *Butterworth* filter as the unseen filter. The filters are summarized in Table I, and their usage during evaluation is detailed in Section III-D. A graphical overview of their different frequency magnitude responses is shown in Fig. 2.

TABLE I
THE TYPES AND ORDERS OF THE LOW-PASS FILTERS USED, UNDER TWO DIFFERENT TRAINING SETTINGS, *SINGLE-FILTER* (NO DATA AUGMENTATION) AND *MULTI-FILTER* (DATA AUGMENTATION)

	Single-filter (No data augmentation)	Multi-filter (Data augmentation)
Training Validation with seen filter(s)	Chebyshev-1, 6	Chebyshev-1, 6 Chebyshev-1, 8 Chebyshev-1, 10 Chebyshev-1, 12 Bessel, 6 Bessel, 12 Elliptic, 6 Elliptic, 12
Validation with unseen filter Testing with unseen filter	Butterworth, 6	Butterworth, 6
Testing with seen filter	Chebyshev-1, 6	Chebyshev-1, 6

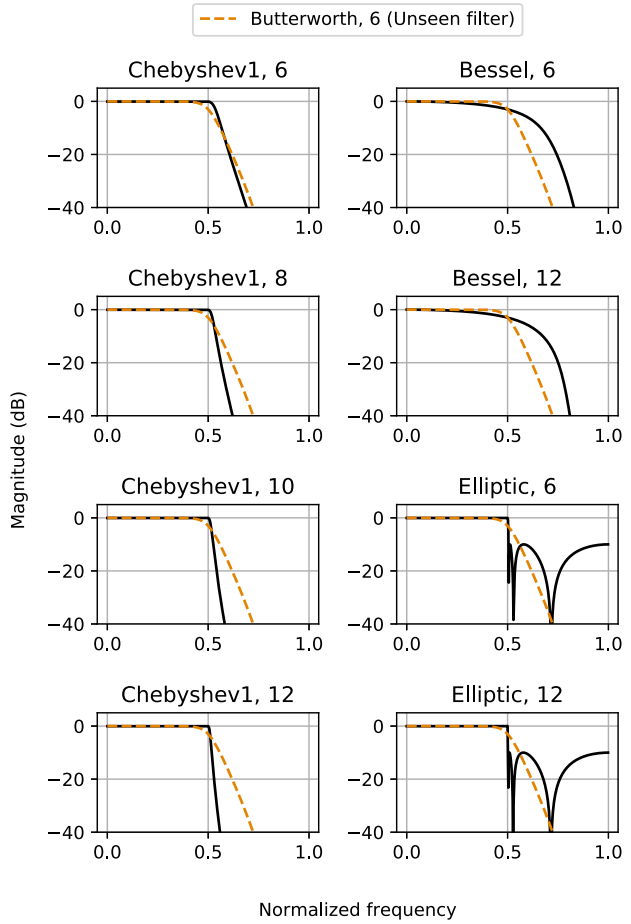


Fig. 2. Frequency responses of the training filters. The frequency response of the unseen filter, 6th order Butterworth is superimposed on each plot.

C. Dataset

Machine learning approaches to bandwidth extension formulate the problem via the use of datasets that contain both full-bandwidth (high-quality) and band-limited (low-quality) versions of each audio signal. A straightforward way to construct these pairs of samples is to obtain a high-quality dataset and then to low-pass filter it. Even though there are many musical audio datasets, especially within the music information retrieval community, many of them are collated from diverse sources (including researchers' personal audio collections) and often contain audio content has been compressed (e.g. via lossy MP3/AAC encoding), hence they are not strictly full-bandwidth nor easily reproducible.

Other than the need for full-bandwidth musical audio content, our proposed approach is intended to be agnostic to musical style. To this end, any uncompressed full-bandwidth musical content could be used as training material, however, to allow reproducibility, we select the following two publicly available datasets, which contain full-bandwidth, stereo, and multi-track musical audio: MedleyDB (version 2.0) [58] and DSD100 [59]. In each dataset, the audio content is sampled at 44100 Hz, with a bandwidth of 22050 Hz.

MedleyDB consists of 121 songs, while DSD100 has two splits for training and testing, each containing 50 songs. Given the inclusion of isolated multi-track stems, both datasets have found high uptake in music mixing and audio source separation research. However, in this work, we seek to address bandwidth extension for multi-instrument music as opposed to isolated single instruments, and thus we retain only the stereo mixes of each song. To create band-limited input samples, we apply a low-pass filter with a fixed cut-off of 11025 Hz, i.e., half the bandwidth of the original. Dataset samples contain values ranging from -1 to 1 and we haven't performed any additional pre-processing, e.g., loudness normalization.

The DSD100 test split is used for testing, the last 8 songs of DSD100 training split are used for validation, with all remaining songs of DSD100 training split plus the entire MedleyDB dataset are used for training. On this basis, the training, validation, and testing sets are all disjoint.

D. Evaluation

1) Metrics: To measure the overall distortion of the outputs, we use the well-established signal-to-noise ratio (SNR):

$$\mathrm{SNR}(x, \hat{x}) = 10 \log_{10} \frac{||x||_2^2}{||x - \hat{x}||_2^2} \tag{1}$$

where x is the reference signal and $\hat{x}$ is its approximation. While calculating the 2-norms, the signals are used in their stereo forms. In the specific context of our work, we consider SNR to be an appropriate choice to investigate overfitting since our models are trained with the mean-squared loss, and minimizing it corresponds to maximizing SNR.

To provide additional insight into performance, we evaluate the perceptual quality of the output audio samples, using the VGG distance, as used recently by Li *et al.* for the evaluation of music enhancement [29]. The VGG distance between two audio samples is defined as the distance between their embeddings created by the *VGGish* network pre-trained on audio classification [28]. A recent work on speech processing shows that the distance between deep embeddings correlates better to human evaluation,

compared to hand-crafted metrics such as Perceptual Evaluation of Speech Quality (PESQ) [60] and the Virtual Speech Quality Objective Listener (ViSQOL) [61], across various audio enhancement tasks including bandwidth extension [62]. The *VGGish* embeddings are also used in measuring the Fréchet Audio Distance (FAD), a state-of-the-art evaluation method to assess the perceptual quality of a collection of output samples [63]. However, because FAD is used to compare two collections rather than individual audio signals, it is not applicable in our case.

To obtain the VGG embeddings, we used the *VGGish* network's open-source implementation [64]. We used the default parameters except setting the sampling frequency to 44100 Hz and the maximum frequency to 22050 Hz. In contrast to the SNR calculation, the reference implementation downmixes the stereo signals to mono before calculating the VGG embeddings. After post-processing, the embeddings take values from 0 to 255. Similar to Manocha *et al.* [62], we employ the mean absolute distance to define the VGG distance as:

$$\mathrm{VGG}(x, \hat{x}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

where x is the reference signal, $\hat{x}$ is its approximation; y and $\hat{y}$ are their embeddings, respectively. n is the size of the embedding tensors, which depends on the length of x.

Given the need to make a large number of objective measurements throughout the training and testing (as detailed in Section IV), we do not pursue any subjective listening experiment and leave this as a topic for future work.

2) Testing: To assess the overall performance of our models, we perform testing once, at the end of the training. The test split of the DSD100 dataset is reserved for our testing stage. Due to GPU memory limitations, our networks cannot process full-length songs in a single forward pass, hence they process non-overlapping chunks of audio and the outputs are later concatenated to create full-length output songs. For both VGG distance and SNR, we calculate them at the song level first, based on these full-length songs, and then take the mean over the data split to obtain the final test values.

To evaluate filter generalization, we perform two tests for each model, using seen and unseen filters. As summarized in Table I, the 6th order Butterworth filter is selected as the unseen filter, as it is not used in any training setting. The seen filter only includes 6th order Chebyshev-1, as this is the only filter common to both single and multi-filter training settings.

3) Validation: To observe generalization or overfitting throughout training, we perform validation repeatedly, where we measure the output SNR once every 2500 training iterations. We perform validation on 8 s audio excerpts, starting from the 8th second of each song, for only 8 songs. These 8 songs correspond to the last 8 of the DSD100 training split. Since the validation is performed repeatedly throughout training, we keep the validation set sample size small. We believe that this small sample size is sufficient, because validation is only used to observe the progress of training, and the final performance evaluation is done in the testing stage. The final validation SNR is obtained by first calculating it over each 8 s, and then taking the mean over the validation songs.

Similar to testing, the validation is also performed twice, using seen and unseen filters. Validation with the unseen filter uses the 6th order Butterworth filter, as in testing. Because validation with the seen filter(s) is done to observe the training progress of each model and not to compare different models, the filters employed are the same as those in the corresponding training setting. As seen in Table I, in the single-filter setting, validation with the seen filter only has the Chebyshev-1 filter, and in the multi-filter setting, it uses all eight training filters, with each assigned to processing a different song in the validation data split.

E. Implementation Details

We built and trained our models using the Pytorch framework [65] and a single Nvidia GeForce GTX 1080 Ti GPU. The model weights are initialized randomly with values drawn from the normal distribution with zero mean and unit variance. The batch size is 8. We use the Adam optimizer [66] with an initial learning rate of 5e-4, and with beta values 0.9 and 0.999. The learning rate is halved when the training loss reaches a plateau. We record the average training loss every 2500 iterations, and consider a plateau to correspond to no decrease in loss for 5 such consecutive measurements. Training samples are created by first randomly picking an audio file from the training dataset and then, at a random location in the audio file, extracting a chunk of stereo audio, with a length of 8192 samples, corresponding to 186 milliseconds. However, since all our models are fully-convolutional, they can process audio signals with arbitrary lengths. We train our models until convergence and for testing we use the model weights taken from the conclusion of the training.

IV. Results

A. Validation Data

Figure 3 provides a high-level overview of the performance of all of the different models and training schemes, with the SNR as a function of the training iterations. While the horizontal dashed lines indicate a baseline of input SNR levels, the rest of the lines denote output SNRs for both validation settings. The SNR levels of the input validation with seen filter(s) are different for the experiments with data augmentation since a different number of training filters are used as summarized in Table I, and as shown in Fig. 2 their differing frequency responses naturally lead to different baseline SNRs.

Examining the first row of Fig. 3 we see that for both networks, when the input filter is known, then large improvements in SNR over the baseline are possible. However, contrasting the U-Net with the ResNet, the performance with the unseen filter is markedly different. For the U-Net the output SNR converges to the baseline, but for the ResNet, performance degrades as training continues. In this way, we see quite clear evidence of a lack of filter generalization in both models.

Moving to the second row, where training includes data augmentation, we observe a different pattern, where both networks

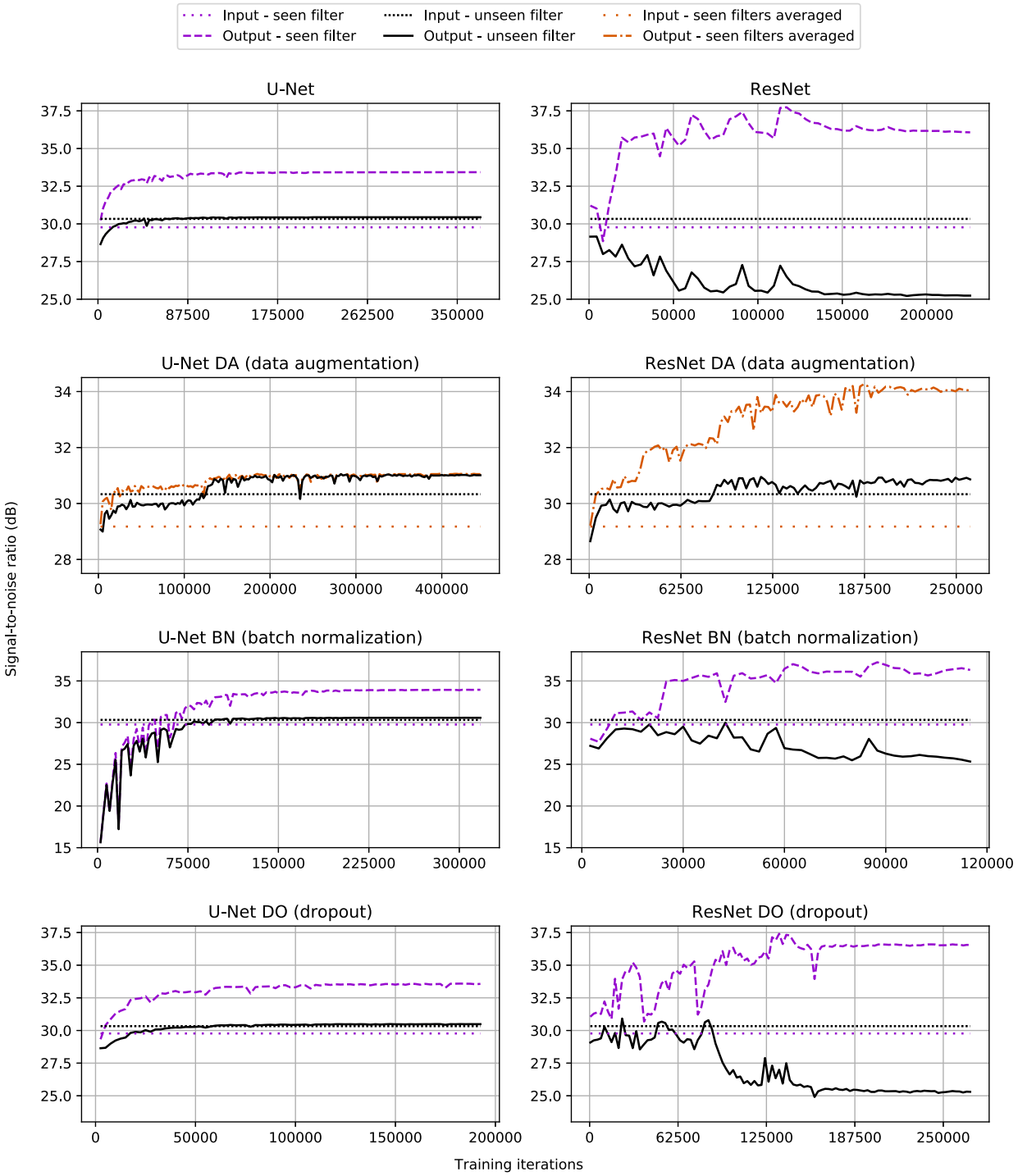


Fig. 3. Validation performance of our models throughout training. The input and output SNR levels are measured by comparing input and model output samples against the ground-truth. Since the inputs are not affected by training, their SNR levels stay constant throughout and constitute the baselines. The seen and unseen filters are detailed in Table I. For the data augmentation experiments, there are multiple seen filters, and the SNR levels are computed by taking the average across multiple filters.

improve upon the baseline SNR for the unseen filter. Contrasting the U-Net and ResNet, we see that the ResNet offers a greater improvement upon the set of seen filters than the U-Net, albeit for approximately the same number of parameters.

Inspection of the third and fourth rows which include the two regularization techniques, we can observe a similar pattern to the first row, where the U-Net again converges to the input SNR, and the ResNet results in a lower SNR than the input. In summary, we see that for both networks, it is only when training with data augmentation that we are able to find any clearly visible improvement in SNR over the input for the unseen filter condition.

B. *Testing Data*

As described in Section III-D3, the validation dataset is small, and the results shown in Fig. 3 are calculated and averaged across short excerpts of 8 s in duration. In Table II, we present the performance of our models on the testing data, which now includes the measurement of the SNR and the VGG distance

TABLE II
OUTPUT SIGNAL-TO-NOISE RATIO (SNR) AND ABSOLUTE VGG DISTANCES (VGG) ON THE TEST DATASET, AND THEIR IMPROVEMENTS WITH RESPECT TO THE INPUTS. FOR SNR, ΔSNR AND $-\Delta$VGG HIGHER IS BETTER AND FOR VGG LOWER IS BETTER. DA, BN, AND DO CORRESPOND TO DATA AUGMENTATION, BATCH NORMALIZATION, AND DROPOUT, RESPECTIVELY. THE VALUE RANGE OF THE VGG EMBEDDINGS AND THE VGG DISTANCES IS 0 TO 255

Filter	Experiment	SNR	ΔSNR	VGG	$-\Delta$VGG
Chebyshev1$-$6 (seen filter)	Input	27.86		46.55	
	U-Net	30.34	+2.47	41.04	+5.51
	U-Net DA	29.78	+1.91	44.29	+2.26
	U-Net BN	30.90	+3.03	41.52	+5.03
	U-Net DO	30.49	+2.62	41.51	+5.04
	ResNet	34.94	+7.08	39.02	+7.53
	ResNet DA	30.48	+2.62	40.11	+6.43
	ResNet BN	34.37	+6.50	39.41	+7.14
	ResNet DO	**35.27**	**+7.41**	**37.23**	**+9.32**
Butterworth$-$6 (unseen filter)	Input	27.37		47.11	
	U-Net	28.55	+1.18	41.90	+5.21
	U-Net DA	29.00	+1.63	44.80	+2.31
	U-Net BN	28.77	+1.40	42.06	+5.06
	U-Net DO	28.62	+1.24	42.34	+4.78
	ResNet	21.96	$-$5.41	47.12	$-$0.01
	ResNet DA	**29.16**	**+1.78**	**40.52**	**+6.59**
	ResNet BN	23.23	$-$4.14	46.38	+0.73
	ResNet DO	22.10	$-$5.27	46.15	+0.96

as a perceptual measure, across the entire duration of the test dataset. When tested with the seen filter, the ResNet models without data augmentation outperform all variants of U-Net by at least 4 dB, achieving more than a 7 dB improvement over the input SNR. The best performing model is ResNet with dropout, improving upon the input SNR by 7.4 dB. We also observe that the inclusion of data augmentation reduces performance when evaluated using the seen filter.

When tested with the unseen filter, the two best performing models use our proposed data augmentation method. Here, the ResNet variants without data augmentation produce output SNR levels well below those of the input. The addition of data augmentation improves the performance of both the baseline U-Net and ResNet. Although this improvement is marginal for the U-Net, at 0.45 dB, for the ResNet, we observe a much larger improvement of 7.2 dB. In testing with the unseen filter, the best performing model is the ResNet with data augmentation, which improves upon the input SNR by 1.8 dB.

Considering the VGG distances, the results of the U-Net variants do not change much across different filters. Compared to the seen filter setting, the ResNet variants without data augmentation exhibit worse results with the unseen filter, however, these values are very close to the input value, hence the filter overfitting in terms of the VGG distance is not as severe as the SNR. For the unseen filter setting, while the incorporation of data augmentation worsens the VGG distance by 2.8 for the U-Net, it produces a much larger improvement of 6.6 for the ResNet, making ResNet with data augmentation the best performing model in terms of VGG distance and SNR.

Quantitative results for each test song, along with three audio excerpts can be found at the following link.[1]

[1][Online]. Available: https://serkansulun.com/bwe

TABLE III
OUTPUT SIGNAL-TO-NOISE RATIO (SNR) OF OUR BASELINE MODELS, WITHOUT ANY REGULARIZATION ON THE TRAINING AND TESTING DATA SPLITS SEPARATELY, AND THEIR IMPROVEMENTS WITH RESPECT TO THE INPUT. THE INPUTS ARE CREATED USING THE LOW-PASS FILTER WHICH WAS ALSO USED DURING TRAINING (THE SEEN FILTER, 6TH ORDER CHEBYSHEV-1).

Data split	Experiment	SNR	ΔSNR
Training	Input	25.99	
	U-Net	28.34	+2.35
	ResNet	33.00	+7.01
Testing	Input	27.86	
	U-Net	30.34	+2.48
	ResNet	34.94	+7.08

C. Sample Overfitting

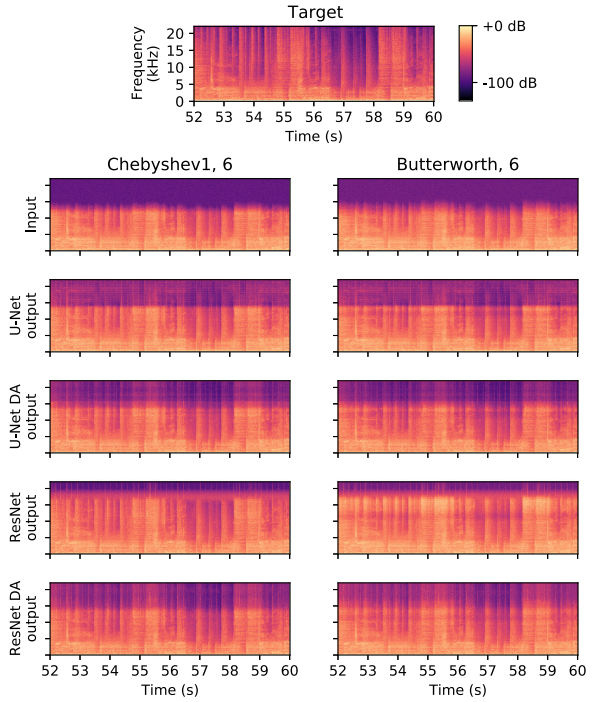
In Table III we present the SNR performance of our baseline models, without any regularization method, on the training and testing data splits separately, and evaluated on all samples in the data splits, across their full duration. To infer whether sample overfitting is occurring (i.e., that the networks are in some sense memorizing the audio content of the training data) we use the seen filter, the 6th order Chebyshev-1. For both the baseline U-Net and ResNet, between training and testing data splits, the SNR improvement over the input is very similar suggesting no overfitting to the audio samples themselves.

D. Visualization of Bandwidth Extension

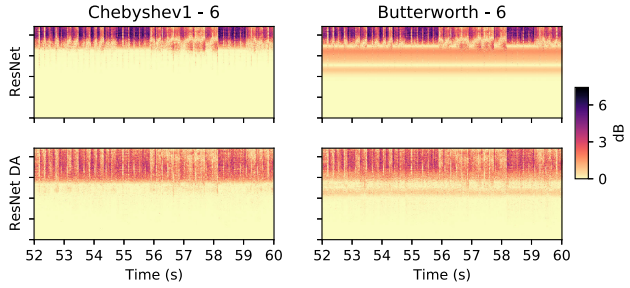
While our proposed method operates entirely in the time-domain, we provide a graphical overview of the outputs of the two networks contrasting the baseline versions with the inclusion of data augmentation for both seen and unseen filters. To this end, we illustrate the spectrograms of one audio excerpt from the test set under each of these conditions in Fig. 4a. The inspection of the figure reveals quite different behavior of the U-Net compared to the ResNet. In general, we can observe more prominent high-frequency information in the output of the ResNet. Of particular note, is the frequency region between approximately 12-17 kHz for the baseline ResNet, and the unseen Butterworth filter, which, contrasting with the target, appears to have "over-enhanced" this region. By contrast, once the data augmentation is included, this high-frequency boosting is no longer evident. To emphasize this phenomenon further, in Fig. 4 b we display the absolute difference with respect to the target spectrogram, for the baseline ResNet and ResNet with data augmentation. For the unseen Butterworth filter, in the upper half of the spectrogram, the absolute difference of the ResNet with data augmentation is much smoother compared to the baseline ResNet. In this visual representation, we can clearly observe that under all conditions the lower part of the absolute difference spectrogram is essentially unchanged, which reflects the direct additive connection of the input to the output in the network architectures.

E. U-Net vs ResNet: Model Comparison

As stipulated in Section III-A, we allow both the U-Net and ResNet to have a similar number of parameters. However, we informally observed a distinct difference in training time. In Table IV, we show several objective properties of these



(a) Spectrograms of sample audio segments.



(b) Absolute difference with respect to the target spectrogram. The colormap is inverted for better visibility.

Fig. 4. Spectrograms and their absolute errors of the sample audio segments. All plots share the axes used in the target (top) plot. Titles per columns denote the type and the order of the filters used. Spectrograms are created using a 1024-sample Hann window with 50% overlap. The audio excerpt is taken from our test set: *DSD100/Mixtures/Test/034 - Secretariat - Over The Top/mixture.wav*

TABLE IV
NUMBER OF PARAMETERS, NUMBER OF MULTIPLY-ACCUMULATE OPERATIONS (MACS), AND RUNTIMES OF OUR MODELS. THE NUMBER OF MACS ROUGHLY CORRESPONDS TO HALF OF THE NUMBER OF FLOATING-POINT OPERATIONS (FLOP). RUNTIME RATE IS THE TIME SPENT IN SECONDS, TO PROCESS A SIGNAL WITH A LENGTH OF ONE SECOND, DURING TESTING, I.E., A FORWARD PASS WHERE NO GRADIENTS ARE CALCULATED

Model	Number of parameters	Number of MACs	Runtime rate
U-Net	56.4M	415.3G	0.14
ResNet	55.1M	3609.4G	1.06

networks, namely the number of parameters, the number of multiply-accumulate operations (MACs), and the runtimes of our baseline models. Therefore, while both models have roughly the same number of parameters, we see that the U-Net has a much lower runtime and fewer MACs. This is due to its autoencoder-like shape, in which the convolutional layers with more channels are near the bottleneck of the network, where the spatial activation size is the smallest, effectively reducing the number of MACs and the runtime. Looking again at Fig. 3, we can speculate that the ResNet has greater representation power than the U-Net, as shown by its ability to better model multiple known filters than the U-Net, albeit at the cost of slower training and inference.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we have raised the issue of filter generalization for deep neural networks applied to musical audio bandwidth extension. Contrary to many problems for which deep learning is used, we do not find any evidence of overfitting to audio samples themselves (i.e. the training data), but rather, we observe a clear trend for state-of-the-art DNNs to overfit to filter shapes. When these DNNs are presented with audio samples that have been pre-processed with low-pass filters that do not match the single training filter, then the scope for meaningful extension of the bandwidth is drastically reduced. Furthermore, the use of widely adopted regularization layers such as batch normalization and dropout fall short in alleviating this problem. Looking to the wider context and long-term goal of musical audio bandwidth extension for audio enhancement, we believe that filter overfitting is a critical issue worthy of continued focus.

To address the filter overfitting issue, we have proposed a novel data augmentation approach, which uses multiple filters at the time of training. Our results demonstrate that without data augmentation, filter overfitting increases as training progresses, whereas including data augmentation is a promising step towards achieving filter generalization. While the improvement in generalization for the U-Net is quite small, a more pronounced effect can be observed for the ResNet, which retains high performance across multiple seen training filters. It is particularly noteworthy that the ResNet variants without data augmentation produce very poor results when tested with an unseen filter, with output quality well below that of the input. In this way, the incorporation of data augmentation was the only means to achieve SNR levels that are above the input.

In addition to the primary findings concerning filter generalization, this is, to the best of our knowledge, the first comparison between U-Nets and ResNets in the field of audio processing, and perhaps the first-ever comparison of these approaches given a similar number of parameters. Examining the results of testing with the seen filter, we observe that the baseline ResNet outperforms the baseline U-Net by a large margin. However, when tested with the unseen filter, the baseline ResNet performs the worst.

We argue that the ResNet has more representation power than the U-Net because while the U-Net reduces the spatial activation sizes in its downsampling blocks, the ResNet keeps the spatial activation sizes constant, starting from its input until its output, thus minimizing the loss of information. Even though the networks have the same number of parameters, we can quantify this higher representation power by comparing the number of MACs. This higher representation power results in the ResNet performing much better in tests with the seen filter, while demonstrating much higher levels of filter overfitting when there

is no data augmentation. We show that using the proposed data augmentation method, this powerful network can be successfully regularized, and achieves the best SNR when tested with the unseen filter.

However, if trained without the proposed data augmentation method and tested using an unseen filter, the U-Net has less tendency to overfit, making it a more robust network compared to the ResNet in this scenario. Furthermore, while we chose to keep the number of parameters within the two models roughly equal, we note that compared to the ResNet, the U-Net is 7.5 times faster and does nearly 9 times fewer MACs. In this way, the U-Net may be a preferred architecture for real-time streaming applications.

Considering our findings in the broader context of audio enhancement and the potential application to archive recordings, we recognize that low-pass filtering alone is by no means sufficient to model the multiple types of signal degradation that can occur. If we wish these trained models to be effective outside of the rather controlled conditions demonstrated here, more work must be undertaken to expand the vocabulary of sound transformations to represent signal degradations including reverberation, wow and flutter, additive noise, and clipping. In this light, the ability of the ResNet with data augmentation to contend with multiple seen filters holds significant promise for a more powerful model to be developed in the future.

A further limitation of our current work is the reliance on SNR and the VGG distance as the indicators of performance. In future work, we consider it of paramount importance to conduct listening experiments to investigate the possible correlations between the subjective evaluations and quantitative perceptual metrics, and to explore models that can improve the perceptual quality such as GANs. Looking beyond the assessment of the perceptual quality of the bandwidth extension, we also seek to investigate listener enjoyment of enhanced archive recordings. Finally, we recognize the potential application of our work on filter generalization to be applied to other types of audio signals, in particular, speech.

REFERENCES

[1] S. Godsill, P. Rayner, and O. Cappé, "Digital audio restoration," in *Applications of Digital Signal Processing to Audio and Acoustics*. Springer, 2002, pp. 133–194.

[2] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "Audio inpainting," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 922–932, Mar. 2012.

[3] N. Perraudin, N. Holighaus, P. Majdak, and P. Balazs, "Inpainting of long audio segments with similarity graphs," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 26, no. 6, pp. 1083–1094, Jun. 2018.

[4] Y. M. Cheng, D. O'Shaughnessy, and P. Mermelstein, "Statistical recovery of wideband speech from narrowband speech," *IEEE Trans. Speech Audio Process.*, vol. 2, no. 4, pp. 544–548, Oct. 1994.

[5] Y. Yoshida and M. Abe, "An algorithm to reconstruct wideband speech from narrowband speech based on codebook mapping," in *Proc. 3rd Int. Conf. Spoken Lang. Process.*, 1994, pp. 1591–1594.

[6] J. Epps and W. H. Holmes, "A new technique for wideband enhancement of coded narrowband speech," in *Proc. IEEE Workshop Speech Coding Proc. Model, Coders, Error Criteria*, 1999, pp. 174–176.

[7] Y. Nakatoh, M. Tsushima, and T. Norimatsu, "Generation of broadband speech from narrowband speech using piecewise linear mapping," in *Proc. 50th Eur. Conf. Speech Commun. Technol., EUROSPEECH*, vol. 3, 1997, pp. 1643–1646.

[8] S. Chennoukh, A. Gerrits, G. Miet, and R. Sluijter, "Speech enhancement via frequency bandwidth extension using line spectral frequencies," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 1, 2001, pp. 665–668.

[9] K.-Y. Park and H. S. Kim, "Narrowband to wideband conversion of speech using GMM based transformation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process*, 2000, vol. 3, pp. 1843–1846.

[10] A. H. Nour-Eldin and P. Kabal, "Mel-frequency cepstral coefficient-based bandwidth extension of narrowband speech," in *Proc. INTERSPEECH, 9th Annu. Conf. Int. Speech Commun. Assoc.*, 2008, pp. 53–56.

[11] P. Jax and P. Vary, "On artificial bandwidth extension of telephone speech," *Signal Process.*, vol. 83, no. 8, pp. 1707–1719, 2003.

[12] P. Bauer and T. Fingscheidt, "An HMM-based artificial bandwidth extension evaluated by cross-language training and test," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2008, pp. 4589–4592.

[13] G.-B. Song and P. Martynovich, "A study of HMM-based bandwidth extension of speech signals," *Signal Process.*, vol. 89, no. 10, pp. 2036–2044, Oct. 2009.

[14] D. Bansal, B. Raj, and P. Smaragdis, "Bandwidth expansion of narrowband speech using non-negative matrix factorization," in *Proc. INTERSPEECH 2005 - Eurospeech, 9th Eur. Conf. Speech Commun. Technol.*, 2005, pp. 1505–1508.

[15] D. L. Sun and R. Mazumder, "Non-negative matrix completion for bandwidth extension: A convex optimization approach," in *Proc. IEEE Int. Workshop Mach. Learn Signal Process. (MLSP)*, 2013, pp. 1–6.

[16] B. Iser and G. Schmidt, "Neural networks versus codebooks in an application for bandwidth extension of speech signals," in *Proc. 8th Eur. Conf. Speech Commun. Technol., EUROSPEECH*, 2003, pp. 565–568.

[17] J. Kontio, L. Laaksonen, and P. Alku, "Neural network-based artificial bandwidth expansion of speech," *IEEE Trans. Audio, Speech Lang. Process.*, vol. 15, no. 3, pp. 873–881, Mar. 2007.

[18] K. Li and C.-H. Lee, "A deep neural network approach to speech bandwidth expansion," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2015, pp. 4395–4399.

[19] Y. Wang, S. Zhao, W. Liu, M. Li, and J. Kuang, "Speech bandwidth expansion based on deep neural networks," in *Proc. INTERSPEECH 2015, 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015.

[20] K. Li, Z. Huang, Y. Xu, and C.-H. Lee, "DNN-based speech bandwidth expansion and its application to adding high-frequency missing features for automatic speech recognition of narrowband speech," in *Proc. INTERSPEECH, 16th Annu. Conf. Int. Speech Commun. Assoc.*, 2015, pp. 2578–2582.

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Image Comput. Comput.-Assisted Intervention*, 2015, pp. 234–241.

[22] V. Kuleshov, S. Z. Enam, and S. Ermon, "Audio super resolution using neural networks," in *Proc. 5th Int. Conf. Learn. Represent.*, 2017.

[23] T. Y. Lim, R. A. Yeh, Y. Xu, M. N. Do, and M. Hasegawa-Johnson, "Time-frequency networks for audio super-resolution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 646–650.

[24] H. Wang and D. Wang, "Time-frequency loss for CNN based speech super-resolution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 861–865.

[25] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[26] S. Kim and V. Sathe, "Bandwidth extension on raw audio via generative adversarial networks," 2019, *arXiv:1903.09027*.

[27] J. Sautter, F. Faubel, M. Buck, and G. Schmidt, "Artificial bandwidth extension using a conditional generative adversarial network with discriminative training," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 7005–7009.

[28] S. Hershey *et al.*, "CNN architectures for large-scale audio classification," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2017, pp. 131–135.

[29] Y. Li, B. Gfeller, M. Tagliasacchi, and D. Roblek, "Learning to denoise historical music," in *Proc. 21st Int. Soc. Music Inf. Retrieval Conf.*, 2020.

[30] B. Adlam, C. Weill, and A. Kapoor, "Investigating under and overfitting in Wasserstein generative adversarial networks," 2019, *arXiv:1910.14137*.

[31] Q. Xu, G. Huang, Y. Yuan, C. Guo, Y. Sun, F. Wu, and K. Weinberger, "An empirical study on evaluation metrics of generative adversarial networks," 2018, *arXiv:1806.07755*.

[32] S. Sulun, "Deep learned frame prediction for video compression," 2018, *arXiv:1811.10946*.

[33] Y. Fan *et al.*, "Balanced two-stage residual networks for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 161–168.
[34] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2017, pp. 1132–1140.
[35] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 2472–2481.
[36] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.,* 2015, vol. 37, pp. 448–456.
[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[38] J. Ming, P. Jancovic, P. Hanna, and D. Stewart, "Modeling the mixtures of known noise and unknown unexpected noise for robust speech recognition," in *EUROSPEECH Scandinavia, 7th Eur. Conf. Speech Commun. Technol.*, 2001, pp. 1111–1114.
[39] S. Mirsamadi and J. H. Hansen, "Multi-domain adversarial training of neural network acoustic models for distant speech recognition," *Speech Commun.*, vol. 106, pp. 21–30, Jan. 2019.
[40] J. Rajnoha, "Multi-condition training for unknown environment adaptation in robust ASR under real conditions," *Acta Polytechnica*, vol. 49, no. 2, pp. 3–7, 2009.
[41] S. Zhang, M. Lei, B. Ma, and L. Xie, "Robust audio-visual speech recognition using bimodal DFSMN with multi-condition training and dropout regularization," in *Proc. 9th IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 6570–6574.
[42] Y. Shi, N. Zheng, Y. Kang, and W. Rong, "Speech loss compensation by generative adversarial networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2019, pp. 347–351.
[43] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans Pattern Anal. Mach. Intell*, vol. 38, no. 2, pp. 295–307, Feb. 2016.
[44] A. Jansson, E. Humphrey, N. Montecchio, R. Bittner, A. Kumar, and T. Weyde, "Singing voice separation with deep u-net convolutional networks," in *Proc. 18th Int. Soc. for Music Inf. Retrieval Conf.*, 2017, pp. 23–27.
[45] T. Y. Lim, R. A. Yeh, Y. Xu, M. N. Do, and M. Hasegawa-Johnson, "Time-frequency networks for audio super-resolution," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2018, pp. 646–650.
[46] C. Macartney and T. Weyde, "Improved speech enhancement with the Wave-U-Net," 2018, *arXiv:1811.11307*.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[48] M. Rempfler, S. Kumar, V. Stierle, P. Paulitschke, B. Andres, and B. H. Menze, "Cell lineage tracing in lens-free microscopy videos," in *Med. Image Comput. Comput. Assisted Intervention*, 2017, pp. 3–11.
[49] V. Ghodrati, J. Shao, M. Bydder, Z. Zhou, W. Yin, K.-L. Nguyen, Y. Yang, and P. Hu, "MR image reconstruction using deep learning: evaluation of network structure and loss functions," *Quantitative Imag Med. Surg.*, vol. 9, no. 9, 2019, Art. no. 1516.
[50] S. Wu, J. Xu, Y.-W. Tai, and C.-K. Tang, "Deep high dynamic range imaging with large foreground motions," in *Proc. Eur. Conf. Comput. Vision*, 2018, pp. 117–132.
[51] E. Chiou, F. Giganti, E. Bonet-Carne, S. Punwani, I. Kokkinos, and E. Panagiotaki, "Prostate cancer classification on verdict DW-MRI using convolutional neural networks," in *Proc. Mach. Learning Med. Imag. - 9th Int. Workshop*, 2018, pp. 319–327.
[52] M. Miron and M. E. P. Davies, "High frequency magnitude spectrogram reconstruction for music mixtures using convolutional autoencoders," in *Proc. 21st Int. Conf. Digit. Audio Effects*, 2018, pp. 173–180.
[53] W. Shi *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 1874–1883.
[54] V. Kuleshov, "kuleshov/audio-super-res," Apr. 2020, original-date: 2017-03-13T02:47:00Z. [Online]. Available: https://github.com/kuleshov/audio-super-res
[55] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. Brit. Mach. Vision Conf.*, 2016.
[56] L. Taylor and G. Nitschke, "Improving deep learning with generic data augmentation," in *Proc. IEEE Symp. Ser. Comput. Intell.*, 2018, pp. 1542–1547.
[57] B. McFee, E. J. Humphrey, and J. P. Bello, "A software framework for musical data augmentation," in *Proc. 16th Int. Soc. Music Inf. Retrieval Conf.*, 2015, pp. 248–254.
[58] R. Bittner, J. Wilkins, H. Yip, and J. Bello, "MedleyDB 2.0: New data and a system for sustainable data collection," in *Proc. 17th Int. Soc. Music Inf. Retrieval Conf.*, 2016.
[59] A. Liutkus, F.-R. Stöter, Z. Rafii, D. Kitamura, B. Rivet, N. Ito, N. Ono, and J. Fontecave, "The 2016 signal separation evaluation campaign," in *Proc. Latent Variable Anal. Signal Separation-12th Int. Conf.*, Cham, 2017, pp. 323–332.
[60] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2001, pp. 749–752.
[61] A. Hines, J. Skoglund, A. Kokaram, and N. Harte, "ViSQOL: The virtual speech quality objective listener," in *Proc. IWAENC—Int. Workshop Acoust. Signal Enhancement*, 2012, pp. 1–4.
[62] P. Manocha, A. Finkelstein, R. Zhang, N. J. Bryan, G. J. Mysore, and Z. Jin, "A differentiable perceptual audio metric learned from just noticeable differences," in *Proc. INTERSPEECH, 21st Annu. Conf. Int. Speech Commun. Assoc.*, 2020.
[63] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proc. INTERSPEECH, 20th Annu. Conf. Int. Speech Commun. Assoc.*, 2019, pp. 2350–2354.
[64] Google, "VGGish," Accessed: Sep. 2, 2020. [Online]. Available: https://github.com/tensorflow/models/tree/master/research/audioset/vggish
[65] A. Paszke *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Adv. Neural Inf. Process. Syst. 32*, pp. 8024–8035, 2019.
[66] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent*, 2015.

Serkan Sulun received the B.Sc. degree from Sabanci University, Istanbul, Turkey, in 2014, and the M.Sc. degree from Koc University, Istanbul, Turkey, in 2018, both in electronic engineering. He is currently a Researcher with INESC TEC and is currently working toward the Ph.D. degree in electrical and computer engineering with the University of Porto, Porto, Portugal. His main research interests are deep learning, symbolic music generation, and audio, image and video processing.

Matthew E. P. Davies received the B.Eng. degree in computer systems with electronics from King's College London, U.K., in 2001, and the Ph.D. degree in electronic engineering from Queen Mary University of London (QMUL), London, U.K., in 2007. From 2007 until 2011, he was a Postdoctoral Researcher with the Centre for Digital Music, QMUL. In 2013, he worked in the Media Interaction Group, National Institute of Advanced Industrial Science and Technology. From 2014 to 2019, he coordinated the Sound and Music Computing Group, INESC TEC, and is currently a Researcher with the Centre for Informatics and Systems of the University of Coimbra. His main research interests include music information retrieval, evaluation methodology, and creative music systems.

MOVING IN TIME: COMPUTATIONAL ANALYSIS OF MICROTIMING IN MARACATU DE BAQUE SOLTO

Matthew E. P. Davies [1,4] **Magdalena Fuentes**[2] **João Fonseca**[3]
Luís Aly[3,4] **Marco Jerónimo**[3] **Filippo Bonini Baraldi**[5,6]
[1] University of Coimbra, CISUC, DEI, Portugal
[2] CUSP, MARL, New York University, USA
[3] University of Porto, Faculty of Engineering, Portugal
[4] INESC TEC, Portugal
[5] Ethnomusicology Institute (INET-md), FCSH, Universidade Nova de Lisboa, Portugal
[6] Centre de Recherche en Ethnomusicologie (CREM-LESC), Paris Nanterre University, France
mepdavies@dei.uc.pt

ABSTRACT

"Maracatu de baque solto" is a Carnival performance combining music, poetry, and dance, occurring in the Zona da Mata Norte region of Pernambuco (Northeast Brazil). Maracatu percussive music is strongly repetitive, and is played as loud and as fast as possible. Both from an MIR and ethnomusicological perspective this makes a complex musical scene to analyse and interpret. In this paper we focus on the extraction of microtiming profiles towards the longer term goal of understanding how rhythmic performance in Maracatu is used to promote health and well-being. To conduct this analysis we use a set of recordings acquired with contact microphones which minimise the interference between performers. Our analysis reveals that the microtiming profiles differ substantially from those observed in more widely studied South American music. In particular, we highlight the presence of dynamic microtiming profiles as well as the importance of the choice of timekeeper instrument, which dictates how the performances can be understood. Throughout this work, we emphasize the importance of a multidisciplinary approach in which MIR, audio engineering, and ethnomusicology must interact to provide meaningful insight about this music.

1. INTRODUCTION

"Maracatu de baque solto", also known as "Maracatu rural", is a Carnival performance combining music, poetry, and dance, occurring in the Zona da Mata Norte region of Pernambuco (Northeast Brazil). Most inhabitants of this region, dominated by the sugar cane monoculture, are rural workers with very modest income, who invest most of their time and money to participate in the Carnival "desfile" (parade), taking place every year in Recife, the state capital. More than 100 groups of Maracatu de baque solto, of different sizes (ranging from 15-20 members up to 200 members) are currently active, each with their own headquarters ("sede") which are generally linked to individual families. A Maracatu performance is shown in Fig. 1.

Figure 1: Maracatu de baque solto "Leão de Ouro de Condado" during a Carnival parade in Tupaoca (Pernambuco), Feb. 28th, 2017. Photo credit: Filippo Bonini Baraldi.

Maracatu de baque solto differs from another Carnival performance with a similar name, Maracatu "de baque virado," not only in terms of its musical and choreographic features, but also because it has remained a very local cultural practice. Indeed, while Maracatu de baque virado, like other music from Pernambuco (e.g., forró, coco de roda, ciranda), has recently spread out nationally and internationally, Maracatu de baque solto (hereafter shortened to Maracatu) is only performed within a radius of about 100 km^2 and is strongly linked to the local afro-indigenous spiritual and religious practices, specifically, the jurema-umbanda worship [1]. This local dimension explains why, barring a few exceptions, it remains a largely understudied cultural expression. To the best of our knowledge no in-depth analysis of its music has ever been realised.

Previous field research conducted in Condado, a small city located in the Zona da Mata Norte region, suggests that

the protective function of Maracatu is locally expressed by two key concepts: "consonância" (consonance) and "fechar" (closure) [2, 3]. Consonance points to a particular way of behaving, of dancing together, and of producing sounds. It is associated with an idea of high interpersonal coordination, as opposed to the idea of "desmantelo" (fracture and breaking up). The expression "fechar o Maracatu" (closing the Maracatu) refers to various aesthetic strategies used to protect the individual and the community from the threats that are at stake during Carnival. The percussive nucleus of Maracatu plays as fast and loud as possible, saturating the acoustical space and "filling" each metrical position in order to avoid any silence. To this end, the study of systematic microtiming, i.e., intentional deviations from strict metronomic timing, can be a promising first step towards understanding Maracatu performances.

In December 2019, 13 members of a Maracatu group "Leão de Ouro de Condado" were invited to Lisbon. Several public performances were organised which culminated in a two-part event: a parade in which musicians and dancers moved through the streets, followed by a fixed location, outdoor performance. We focus on the recordings obtained from the latter, and perform an exploratory analysis of the microtiming in the percussionists' performances.

Our research is part of the emerging topic of computational ethnomusicology [4, 5], and falls within a growing body of work examining the role of microtiming and rhythmic structure and its relationship to groove and musical embodiment [6–10]. It also intersects with the existing MIR literature on the analysis of rhythm in South American music [11–16]. In our specific context, we face two prominent, interconnected challenges. In the absence of formal theories about Maracatu, there is little basis on which to pose research questions that computational analysis (e.g., using MIR techniques) could address. From a more practical perspective, the very nature of Maracatu performance: musicians in tight proximity to one another, playing very loudly, and moving between multiple ad-hoc external locations, creates technical difficulties for the high-quality signal acquisition necessary for temporally-precise analysis of musical timing.

Our methodology to address these challenges is to conduct the research from a strongly multidisciplinary perspective. We directly leverage technical expertise in audio engineering for signal acquisition, music signal processing and machine learning for computational rhythm analysis, and ethnomusicology to guide the interpretation of the findings based on long-term field research. By necessity, this leads to an exploratory approach concerning the presence and use of microtiming in Maracatu, where the computational analysis can be used as a means to infer new understanding about the musical practice.

To enable the isolated analysis of the microtiming of each percussionist, we use contact microphones attached to each individual instrument. We adapt a state-of-the-art approach for microtiming analysis [12] and investigate both "within-instrument" microtiming, where onset and beat information are specific to a given instrument, and "between-instrument" microtiming, where a "time-keeping" instrument provides the beat reference. Our main findings demonstrate that the choice of the time-keeper is critical and can change the interpretation of the performance. Furthermore, we observe microtiming profiles that change dynamically within given pieces of music, and between pieces of music. Thus, the notion of a single characteristic microtiming profile appears not to apply to the set of Maracatu recordings under investigation here.

The remainder of this paper is structured as follows. Section 2 provides an overview of the musical structure and instruments of Maracatu. Section 3 describes the signal acquisition process. Section 4 summarizes the microtiming modelling approach, with our main findings presented in Section 5. We conclude the paper in Section 6 with a reflection on the broader impact of the research.

2. MARACATU DE BAQUE SOLTO

Maracatu is a combination of various elements: two to four wind instruments (trumpet and trombone), played by "musicos" (musicians), and a nucleus of five percussionists called the "terno." During performances, the musicos and terno act in close cooperation with a poet (the "mestre de apito"). When the poet improvises short verses about 30 s in duration, the musicians remain silent and the dancers and public remain still. When the poet finishes his verses and blows his whistle ("apito"), the percussionists and the wind musicians play for about the same duration and everybody dances. During this time, the poet prepares his next verses. This alternating pattern remains throughout the performance, which may last up to eight hours.

Maracatu percussive music is highly repetitive, and played as loud and as fast as possible. These features give rise to a very strong euphoria in the dancers and the public. Just a few rhythmical patterns exist in Maracatu and depend on the metrical form that the poet is following: the main genres are "marcha" and "samba," although the samba of Maracatu has nothing to do with the well-known samba music of Brazil. In Maracatu, marcha and samba indicate both the metrical subdivision of the poet's couplets, the rhythm played by the terno, and the melodies played by the musicos. Various melodies may be associated to the marcha pattern and/or samba pattern, depending on the melodic line that the poet chooses to sing his verses.

In Maracatu, the five percussion instruments of the terno, shown in Fig. 2, are: *Bombo* – a bass drum-like instrument played with two sticks, one for each side. We refer to Bombo High as the upper skin, and Bombo Low as the lower. *Gonguê* – an iron instrument comprised of two bells of different pitches. We refer to Gonguê High as the higher, and Gonguê Low as the lower pitched bell. *Porca* – a friction drum, played with a damp cloth holding the stick. *Tarol* – similar to a small snare drum, but thinner. *Mineiro* – a metal tube filled with beads or other small objects, which is shaken to create a rattle-type sound.

Following transcriptions in [17] for both the marcha and samba, the Porca plays a regular quarter note pattern. Likewise for the marcha, the lower bell of the Gonguê has

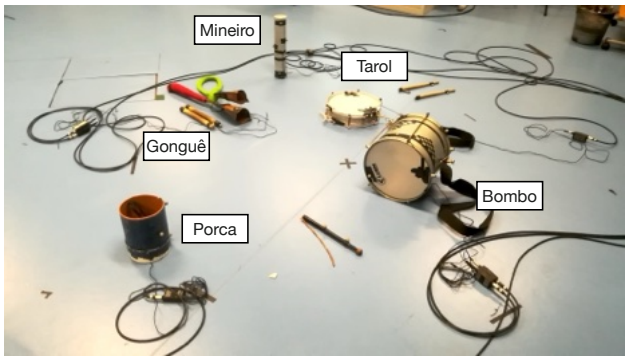


Figure 2: The instruments of the terno (with identifying labels overlaid) including the connection of the contact microphones.

the same pattern. Thus, for marcha we assume that either could provide a time-keeping role.

3. DATA ACQUISITION

In this work, we wished to analyse each percussionist's performance in isolation. While multiple microphone setups have been successfully used to acquire separated audio data for rhythmic and microtiming analysis [18], the physical arrangement of the Maracatu percussionists in a tight circle, together with the very loud playing style, makes this impractical and highly prone to "spillage." In turn, this would make any subsequent annotation and microtiming analysis extremely challenging. A promising alternative is to consider the use of contact microphones.

For this study, we used the Schertler Basik Set universal contact microphone. Each microphone includes a phantom-power adaptor box which delivers a line-level signal, with a 60 Hz–15 kHz frequency response. The sensitivity on the instrument is -34 dB (time-averaged sound level). We found these microphones provided high-quality audio recordings with minimal spillage and distortion.

For the microphone placement, we sought to balance the optimum location for sound capture while minimizing any impact to each musician's playing style. Given the small size of the Schertler pickup (less than 1 cm in diameter), this aspect was relatively straightforward.

We placed two pickups on the Gonguê – one per bell, and two on the Bombo – one per skin. For the remainder, the Tarol, Porca, and Mineiro we used a single pickup. The contact microphones were individually connected to a Motu UltraLite-mk3 Hybrid (USB/Firewire) with nominal gain at the input and no further processing. The recording session consisted of discrete, synchronised tracks, one per microphone, recorded in a Pro Tools 2019 mixing session configured to record at 44.1 kHz, with 16-bit depth.

We used this setup in the fixed location performance, rather than the parade. Over the total performance duration of 48 minutes, we partitioned the performance into 34 individual pieces (i.e., editing out the poetry), with a mean duration of 39.4 s. Of these 34 pieces, 28 were marcha, 5 were samba, and in one piece the percussionists played samba, while the musicos played marcha.

4. MICROTIMING MODEL

In order to undertake any assessment of the microtiming present in recordings, we must obtain precise temporal markers which indicate the note onset positions. Following existing work in microtiming analysis [12, 14] it is necessary to create a reference beat grid against which to compare the locations of performed onsets with quantised beat and/or sub-beat positions. In this way, each beat interval can be assigned a normalised duration of 1, and thus a rhythmic pattern containing four equal sub-divisions would occur at normalised positions 0, 0.25, 0.50, and 0.75. When summarising this information over multiple beats, it is possible to observe systematic microtiming patterns–called microtiming profiles [12]–, where, in the case of Brazilian samba, the third and fourth sub-divisions of the beats have been shown to occur ahead of their quantised position [12, 14]. Note that because the relative position of onsets is normalized with respect to the beat interval, the modelling of microtiming profiles is independent of tempo changes, which allows us to visualise their change over time in a consistent manner. In this section, we first describe the means by which onsets and beat annotations were obtained, followed by the technique used to estimate microtiming profiles through time.

4.1 Onset Annotation

The 34 pieces in the Maracatu performance totalled approximately 22 minutes. Across all 7 channels, this led to a total of 238 contact microphone signals to be analysed. As shown in Fig. 3, even though the separation between channels is mostly very good, some spillage still occurred. This is especially prominent between the bells of the Gonguê which are physically connected. We also observed spillage where one instrument had yet to begin playing and the vibrations carried through the air were picked up thanks to the extremely high sensitivity of the contact microphones.

Given our proposed signal acquisition process using contact microphones, we assumed that largely isolated percussion tracks would be relatively straightforward for a well-known onset detection method using deep neural networks [19]. On this basis, we hoped to be able obtain reliable onset information in a semi-automatic way, where minor corrections (shifts, insertions, and deletions) could be performed. In practice, we found that the rather unusual waveform shapes of the Gonguê, Porca, and Mineiro events created numerous problems for the onset detection system and thus provided little benefit over manual annotation from scratch. Indeed, the recordings of the Mineiro were so challenging to annotate in a precise and consistent way, that we chose not to include them at this stage of our analysis. Ultimately, we selected four instruments: two instruments with time-keeping roles (Porca and Gonguê Low) and two rhythmically expressive instruments in which to observe microtiming (Tarol and Bombo High).

To provide the final onset annotations we followed the

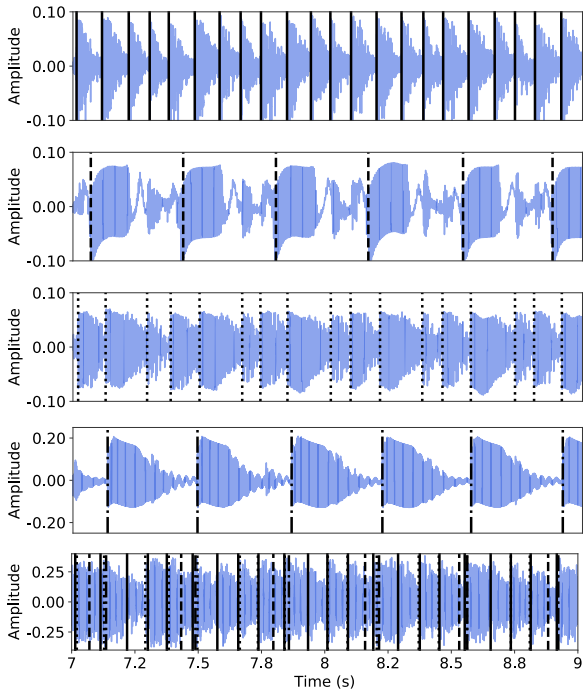


Figure 3: Illustration of signals with annotated onsets in approximately one bar. Top to bottom: Tarol with annotations (solid line); Porca with annotations (dashed line); Bombo High with annotations (dotted line); Gonguê Low with annotations (dash-dotted line); Audio mixture with overlaid onset times of the four instruments.

methodology applied to Brazilian samba [12] and adapted an existing deep neural network [20] and retrained it specific to each instrument using a subset of manually annotations. Even using *instrument-adapted* networks, some manual correction was required, both to contend with issues of temporal localization as well as extra and missed detections. As a coarse indication of the annotation effort, we highlight 51 onsets in just 2 s in the lowest plot of Fig. 3, with $\sim 45,000$ over the four instruments.

4.2 Microtiming Estimation

Our microtiming modelling is based on the approach in [12], which models microtiming profiles per beat as a multi-dimensional variable **m**, where each component m_i with $i = 1, ..., N$, with N the total number of onsets per beat, describes the evolution over time of the relative position of onset i with respect to the beat. The model in [12] estimates the beat and onset likelihoods, and infers the beat positions and associated microtiming profiles jointly using conditional random fields (CRFs). In this work we follow these ideas, but instead of inferring beat and microtiming profiles jointly with a CRF as in [12], we perform the beat and onset inference offline, and then group the onsets and beats using Algorithm 1. The reason for this is two fold: our proposed approach is simpler and computationally cheaper than the CRF approach, with the limitation that it would not be robust in presence of noisy signals or mixtures, which is not the case here. Also, since we are interested in both within-instrument and between-instrument microtiming, we need a flexible model that allows changing the beat reference, which we can do since we compute beats and onsets offline and integrate them afterwards.

Algorithm 1: Microtiming modelling

Input: b, o, τ, r
Output: **m**, t
for *i* ← *1 to len(b)-1* **do**
  $\Delta b \leftarrow b^{(i+1)} - b^{(i)}$;
  $t_{ini} \leftarrow b^{(i)} - \tau \times \Delta b$;
  $t_{end} \leftarrow b^{(i+1)} - \tau \times \Delta b$;
  $o_{beat} \leftarrow o[t_{ini} < o < t_{end}]$;
  if *len*(o_{beat}) $< r$ ***and*** o_{beat} *is not empty* **then**
    $o_{temp} \leftarrow range(0, 1, 1/r) + t_{ini}$;
    for *j* ← *1 to len*(o_{beat}) **do**
      $k_{min} \leftarrow \arg\min_k(|o^{(j)}_{beat} - o^{(k)}_{temp}|)$;
      $o_{fix}[k_{min}] \leftarrow o^{(j)}_{beat}$;
    end
    $o_{beat} \leftarrow \text{interp}(o_{fix}[nan], o_{fix}[\sim nan])$
  else
    continue;
  end
  for $j \leftarrow 2$ *to len*(o_{beat}) **do**
    $v^{(j-1)}_{IOI} \leftarrow o^{(j)}_{beat} - o^{(j-1)}_{beat}$
  end
  $\mathbf{m}^{(i)} \leftarrow v_{IOI}$ / Δb ;
  $t^{(i)} \leftarrow b^{(i)}$
end

Algorithm 1 is structured as follows: for each beat b we obtain the onsets o_{beat} that fall within the beat interval $[t_{ini}, t_{end}]$ by a tolerance given by τ, and check if we have the expected number of onsets (denoted by r in Algorithm 1). If not, we deduce which onsets are missing by comparing the given onsets with a template containing evenly-distributed positions (0.25, 0.5 and 0.75 in the case of three expected onsets). Next, we interpolate the missing onsets for better visualisation of the microtiming pattern. Finally, we obtain the microtiming profile **m** by dividing all inter-onset-intervals by the beat interval length. We exclude beats with more than the expected number of onsets.

Microtiming deviations and their variation across time have been studied in the context of time-keeper instruments (e.g., in Brazilian samba [12], Uruguayan candombe [18], and jazz [21]). While previous approaches focus on within-instrument rhythmic patterns, we also explore the microtiming generated between time-keepers and non-time-keeper instruments. We apply a similar strategy to that in [12] to analyse the profiles visually.

Both existing work and Algorithm 1 assume that there is *one* main rhythmic pattern played during most of the recording. This hypothesis holds in most of the recordings we obtained, however, unlike other examples such as the BRID dataset [22] where it holds to a great extent, in Maracatu, it is not true for samba.

For the analysis of between-instrument microtiming de-

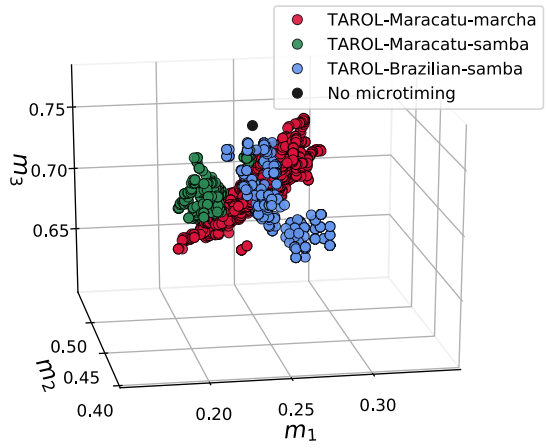


Figure 4: Comparison of Maracatu sub-genres and Tarol (Caixa) in Brazilian samba.

viations, we follow the existing literature about Maracatu [17] and use the Porca and Gonguê Low onsets as beat references for the other instruments since they play the role of time-keepers and are meant to play the beat. For the within-instrument analysis for Tarol and Bombo, we estimated the beats using the *madmom* library [23] [1] , which performed well in the solo tracks, and we adjusted the final beat positions to the closest onset.

5. MICROTIMING ANALYSIS

Marcha vs. samba: Both by listening to the recordings and inspection of the microtiming profiles, we discovered that samba has far greater variation in rhythmic patterns than marcha. In addition, the beat estimation revealed a noticeable difference in tempo, with marcha approximately 165 bpm where as samba was faster at around 180 bpm. Transcriptions from [17] also state that there is greater variation in the notated rhythmic patterns for samba compared to marcha. From the perspective of drawing robust conclusions about microtiming profiles, which, using our approach, rely quite strongly on a consistent rhythmic pattern, we were only able to conduct reliable analysis for Tarol within-instrument microtiming. To enable a high-level comparison, which also includes the use of Tarol (referred to as Caixa in [12]) in Brazilian samba we present a scatter plot of m_1 vs. m_2 vs. m_3 in Fig. 4. Perhaps the most striking observation is that for marcha, the Tarol shows a much wider variation in the m_1 and m_3 dimensions, where as the limited data we obtain for Maracatu samba occupies a tighter cluster, and Brazilian samba varies more prominently over m_2. This indicates a different use of microtiming for Tarol in our recordings both within Maracatu sub-genres and compared to Brazilian samba.

Microtiming profiles in marcha: In Fig. 5 we observe both the within- and between-instrument microtiming analysis for Tarol (left column) and Bombo High (right column) for recording #28. For the between-instrument analysis we use the Gonguê Low and Porca as time-keepers (middle and bottom rows respectively). When comparing between-instrument and within-instrument profiles, we observe one additional trace for both instruments: the deviation at the beat level (which is normalised out for within-instrument analysis). Referring back to Fig. 3 we can see microtiming profiles consistent with a sub-division of the beat into four $1/16$th notes for Tarol, with the second $1/16$th note (m_1) not played for Bombo High.

Looking at the microtiming profiles through time, we see more fluctuation in Tarol compared to Bombo High. In particular for Tarol, the smoothed microtiming profiles move above and below the quantised positions indicating a dynamic use of microtiming. Furthermore, a direct comparison of Tarol with Bombo High illustrates different profiles. Across the entire set of recordings, we found the following median profiles: Tarol $[0.25, 0.47, 0.715]$, and Bombo High $[0.46, 0.69]$.

When contrasting the time-keepers, we observe much greater variation when the Porca beats are used as reference compared to Gonguê Low, including an unexpected downward trend for both Tarol and Bombo High. While not present in all marcha, we observed several similar instances, which can be attributed to the beats of the Porca being played *ahead* of the beat and slowly aligning in phase by the end. Looking again at Fig 3, we see the Porca annotations are earlier than the other instruments consistently by upto 20 ms. A possible explanation may be that the Gonguê is louder and thus takes a more prominent time-keeping role, allowing the Porca to take a more expressive role. Regardless, we assert the importance of analysing the behaviour of the time-keeper instruments.

6. REFLECTIONS

One objective of current ethnomusicological research is to reveal how musicians of different cultures develop strategies for playing together that differ in subtle ways to those in Western culture. These strategies are often implicit, associated to verbal categories that express local views on how music "should sound" in a particular cultural context [25]. These verbal categories often rely on non-musical concepts and metaphors, often related to other sensorial domains (vision, taste, etc.) [26].

Ethnographic field research is a first, necessary step for unveiling subtle strategies of playing together that are at stake in a particular musical culture. Formal analysis of live performances is then needed to understand to what acoustic reality these local concepts refer. To this end, MIR techniques, such as microtiming analysis, can provide innovative solutions for exploring qualities of the music that would otherwise be hard to describe.

In the Zona da Mata region, Carnival is not a simple distraction but rather a ritual involving a complex set of mystical beliefs and social concerns. At this time of the year invisible negative "entidades" (entities) are believed to be more active and even dangerous, and interpersonal relations are marked by feelings of envy and jealousy. Carnival is therefore considered as a threat both for individuals and

[1] We used the model that implements Böck et al. [24] in version 0.16.1.

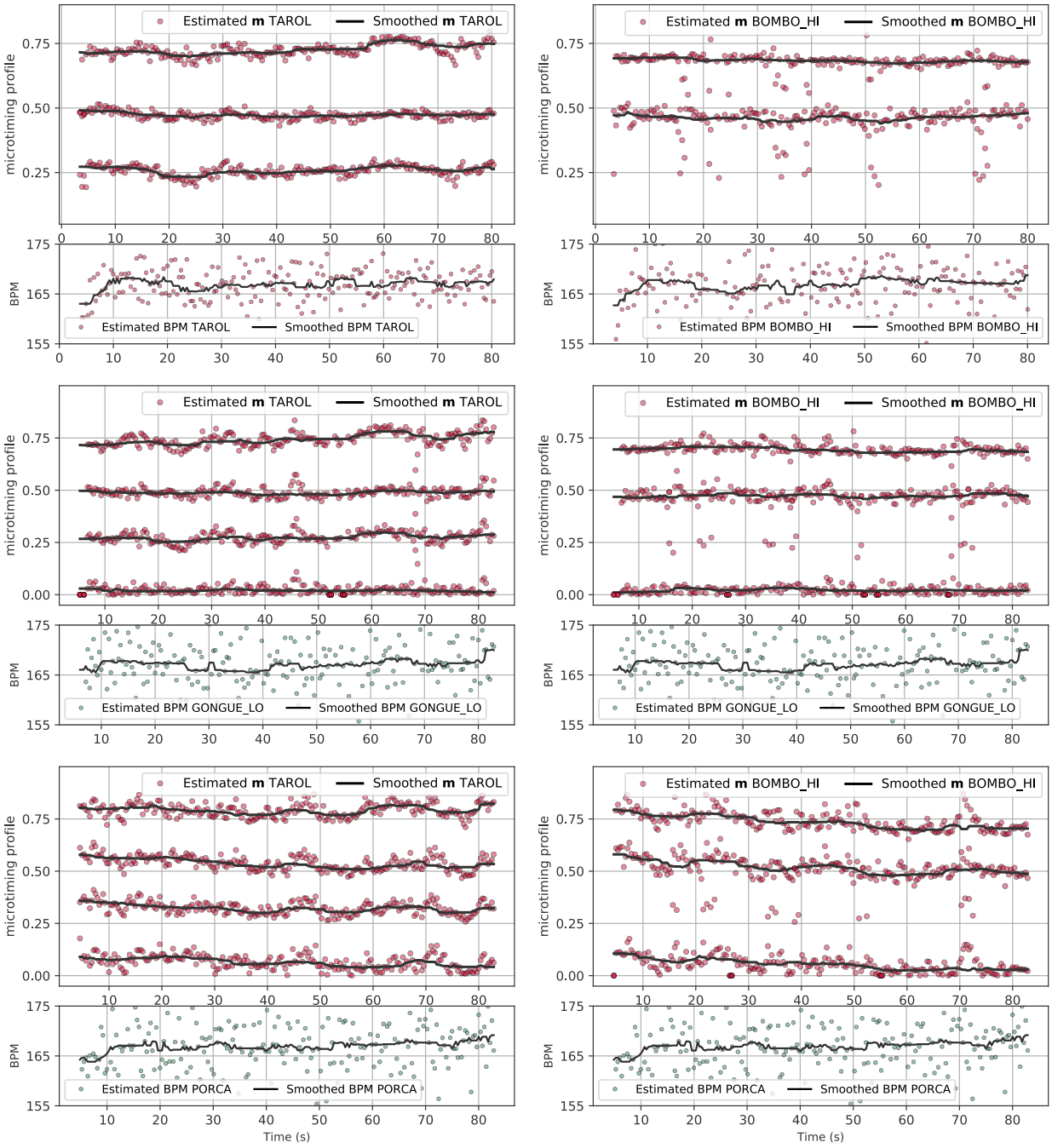


Figure 5: Microtiming profiles in recording #28 for Tarol and Bombo High, left and right respectively. The microtiming estimations use within-instrument, Gonguê Low and Porca beats as reference in top, middle and bottom plots respectively.

the community. Maracatu de baque solto is a performance-ritual that allows people to overcome these risks.

It is important to stress that the Maracatu musicians had never travelled outside of Brazil, and bringing them to Europe was logistically complicated. In addition, the signal acquisition and onset annotation were also non-trivial, meaning several challenges needed to be overcome before even beginning to analyse the microtiming in Maracatu in a computational way. Nevertheless, within the confines of a small dataset, limited to one set of musicians, our preliminary analysis revealed new findings on the use of microtiming, in particular its dynamic nature in Maracatu.

In the long term, our aim is to understand what it means to play in "consonance" and to "close the Maracatu." Both concepts point to subtle manners of producing sounds collectively, that differ form the ones observed in other musical contexts. Since no formal analyses of Maracatu music have been previously realised, this paper is a first attempt to understand how musicians rhythmically interact during a live performance. In future research, we intend to play back the recordings to the musicians to understand if "consonancia" and "closure" can be associated to the specific microtiming profiles highlighted in this paper.

7. ACKNOWLEDGMENTS

This work is supported by Portuguese National Funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project "The Healing and Emotional Power of Music and Dance" (HELP-MD), PTDC/ART-PER/29641/2017.

This work is funded by national funds through the FCT - Foundation for Science and Technology, I.P., within the scope of the project CISUC - UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020 as well as by Portuguese National Funds through the FCT - Foundation for Science and Technology, I.P., under the project IF/01566/2015.

Magdalena Fuentes is a faculty fellow in the NYU Provost's Postdoctoral Fellowship Program at the NYU Center for Urban Science and Progress and Music and Audio Research Laboratory.

We would especially like to thank all 13 members of the "Leão de Ouro de Condado" Maracatu group who visited Lisbon in December 2019; without whom this research would have been impossible.

8. REFERENCES

[1] L. Garrabé, "Les rythmes d'une culture populaire: les politiques du sensible dans le maracatu-de-baque-solto, Pernambuco, Brésil," Ph.D. dissertation, Université Paris 8, 2010, (in French).

[2] M. Acselrad, *Viva Pareia! Corpo, dança e brincadeira no Cavalo-Marinho de Pernambuco*. Recife: Editora Editora Universitária UFPE, 2013, (in Portuguese).

[3] F. Bonini Baraldi, "Inveja e corpo fechado no Maracatu de baque solto pernambucano," *Submitted.* (in Portuguese).

[4] E. Gómez, P. Herrera, and F. Gómez-Martin, "Computational ethnomusicology: perspectives and challenges," *Journal of New Music Research*, vol. 42, no. 2, pp. 111–112, 2013.

[5] G. Tzanetakis, "Computational ethnomusicology: a music information retrieval perspective," in *Proc. of Joint ICMC\SMC Conference*, 2014, pp. 112–117.

[6] M. E. P. Davies, G. Madison, P. Silva, and F. Gouyon, "The effect of microtiming deviations on the perception of groove in short rhythms," *Music Perception: An Interdisciplinary Journal*, vol. 30, no. 5, pp. 497–510, 2013.

[7] A. Hofmann, B. C. Wesolowski, and W. Goebl, "The tight-interlocked rhythm section: Production and perception of synchronisation in jazz trio performance," *Journal of New Music Research*, vol. 46, no. 4, pp. 329–341, 2017.

[8] L. Kilchenmann and O. Senn, "Microtiming in swing and funk affects the body movement behavior of music expert listeners," *Frontiers in psychology*, vol. 6, p. 1232, 2015.

[9] B. Merker, "Groove or swing as distributed rhythmic consonance: introducing the groove matrix," *Frontiers in Human Neuroscience*, vol. 8, 2014.

[10] M. A. Witek, E. F. Clarke, M. Wallentin, M. L. Kringelbach, and P. Vuust, "Syncopation, body-movement and pleasure in groove music," *PloS one*, vol. 9, no. 4, p. e94446, 2014.

[11] T. M. Esparza, J. P. Bello, and E. J. Humphrey, "From genre classification to rhythm similarity: Computational and musicological insights," *Journal of New Music Research*, vol. 44, no. 1, pp. 39–57, 2015.

[12] M. Fuentes, L. S. Maia, M. Rocamora, L. W. P. Biscainho, H. C. Crayencour, S. Essid, and J. P. Bello, "Tracking beats and microtiming in afro-latin american music using conditional random fields and deep learning," in *Proc. of the 20th Intl. Society for Music Information Retrieval Conf.*, 2019, pp. 251–258.

[13] L. S. Maia, M. Fuentes, L. W. P. Biscainho, M. Rocamora, and S. Essid, "SAMBASET: A Dataset of Historical Samba de Enredo Recordings for Computational Music Analysis," in *Proc. of the 20th Intl. Society for Music Information Retrieval Conf.*, 2019, pp. 628–635.

[14] L. Naveda, F. Gouyon, C. Guedes, and M. Leman, "Microtiming patterns and interactions with musical properties in samba music," *Journal of New Music Research*, vol. 40, no. 3, pp. 225–238, 2011.

[15] L. Nunes, M. Rocamora, and L. W. P. Jure, L.and Biscainho, "Beat and downbeat tracking based on rhythmic patterns applied to the Uruguayan Candombe drumming," in *Proc. of the 16th Intl. Society for Music Information Retrieval Conf.*, 2015, pp. 264–270.

[16] M. Rocamora, L. Jure, M. Fuentes, L. S. Maia, and L. W. P. Biscainho, "CARAT: Computer-aided Rhythmic Analysis Toolbox," in *Late-Breaking Demo Session of the 20th Intl. Society for Music Information Retrieval Conf.*, 2019.

[17] C. de Oliveira Santos, T. S. Resende, and P. M. Keays, *Batuque Book: Maracatu Baque Virado e Baque Solto*. Recife: Edição do autor, 2009.

[18] M. Rocamora, "Computational methods for percussion music analysis: The Afro-Uruguayan Candombe drumming as a case study," Ph.D. dissertation, Universidad de la República (Uruguay). Facultad de Ingeniería, 2018.

[19] F. Eyben, S. Böck, B. Schuller, and A. Graves, "Universal onset detection with bidirectional long-short term memory neural networks," in *Proc. of the 11th Intl. Society for Music Information Retrieval Conf.*, 2010, pp. 589–594.

[20] M. E. P. Davies and S. Böck, "Temporal convolutional networks for musical audio beat tracking," in *Proc. of the 27th European Signal Processing Conf.*, 2019.

[21] C. Dittmar, M. Pfleiderer, S. Balke, and M. Müller, "A swingogram representation for tracking micro-rhythmic variation in jazz performances," *Journal of New Music Research*, vol. 47, no. 2, pp. 97–113, 2018.

[22] L. S. Maia *et al.*, "A novel dataset of Brazilian rhythmic instruments and some experiments in computational rhythm analysis," in *AES Latin American Congress of Audio Engineering (AES LAC)*, 2018, pp. 53–60.

[23] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "madmom: a new python audio and music signal processing library," in *24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, Oct. 2016, pp. 1174–1178.

[24] S. Böck and M. Schedl, "Enhanced beat tracking with context-aware neural networks," in *Proc. Int. Conf. Digital Audio Effects*, 2011, pp. 135–139.

[25] N. Fernando and D. Rappoport, Eds., *Cahiers d'ethnomusicologie: Le Goût Musical*, 28, 2015.

[26] F. Bonini Baraldi, E. Bigand, and T. Pozzo, "Measuring Aksak Rhythm and Synchronization in Transylvanian Village Music by Using Motion Capture," *Empirical Musicology Review*, vol. 10, no. 4, pp. 265–291, 2015.

On the Use of Automatic Onset Detection for the Analysis of Maracatu de Baque Solto

João Fonseca, Magdalena Fuentes, Filippo Bonini Baraldi, and Matthew E. P. Davies

Abstract In this work we investigate the use of automatic onset detection techniques to support the ethnomusicological analysis of microtiming in Maracatu de Baque Solto. In order to lay the foundation for a robust and meaningful analysis of Maracatu in terms of its temporal and rhythmic structure, we require extremely precise annotations of note onset positions. However, the nature of Maracatu performance, with percussionists in very close proximity to one another and playing very loud and at high tempo, makes manual annotation extremely challenging. To this end, we orient our work towards minimising the number of corrections necessary by a human annotator, and thus we incorporate explicit knowledge of Maracatu into a computational approach for onset detection. In our evaluation, we explore the divergence between the use of an "off the shelf" state-of-the-art onset detection technique which has been trained to generalise across a wide variety of musical material, with a bespoke approach which specifically targets the instrumentation in Maracatu. We then explore the impact of this approach when visualising microtiming profiles.

J. Fonseca (✉)
Faculty of Engineering, University of Porto, Porto, Portugal
e-mail: up201403802@fe.up.pt

M. Fuentes
CUSP, MARL, New York University, New York City, NY, USA
e-mail: mfuentes@nyu.edu

F. Bonini Baraldi
Ethnomusicology Institute (INET-md), FCSH, Universidade Nova de Lisboa, Lisbon, Portugal
e-mail: fbaraldi@fcsh.unl.pt

Centre de Recherche en Ethnomusicologie (CREM-LESC), Paris Nanterre University, Nanterre, France

M. E. P. Davies
University of Coimbra, CISUC, DEI, and INESC TEC, Coimbra, Portugal
e-mail: mepdavies@dei.uc.pt

L. Correia Castilho et al. (eds.), *Perspectives on Music, Sound and Musicology*,
Current Research in Systematic Musicology 10,
https://doi.org/10.1007/978-3-030-78451-5_11

1 Introduction

The relation among music, health, and culture is currently a main topic of ethnomusicological research [for a review, see Koen et al. (2008)]. Communities from around the globe have developed countless local practices in order to heal people, prevent from illness or enhance well-being by using sounds and music. These practices are generally ritualized and involve a set of symbolic, religious, and emotional meanings that culminate in expressive strategies for playing together that differ from those in Western culture.

During the Carnival season, the inhabitants of Zona da Mata Norte region, in the interior of Pernambuco state (Northeast Brazil), gather around the Maracatu de Baque Solto, a performance-ritual that is perceived to help prevent from threats provoked by invisible entities and to promote health and social equilibrium Bonini Baraldi (2021). The oldest Maracatu de Baque Solto groups were formed toward the end of the 19th century and have been included into the Brazilian National Historical and Artistic Heritage Institute in the Register Book of Forms of Expression since December 2014. The members of these groups are mostly rural workers and sugarcane cutters, with very modest incomes.

Maracatu performances comprise various aesthetical means including costumes, music, dance, and improvised poetry. Music is performed by two to five "músicos" (musicians) playing wind instruments (generally trombone and trumpet), and five percussionists (the "terno") playing the "tarol", "bombo", "gonguê", "porca" and "mineiro" (see Sect. 2). A Maracatu group includes 15–200 masked dancers performing complex collective choreographies ("manobras"), among which the most emblematic character is the "caboclo de lança" (see Fig. 1). Two poets ("mestres do apito") regularly stop the music and the dance by blowing their whistles ("apito"), and improvise short chanted verses in various metrical forms, depending on the moment of the performance. When the poet finishes his verses and blows again his whistle, music and dance resume. This alternating pattern between the poets and the musicians and dancers remains throughout the whole performance, which may last an entire night.

Maracatu musicians and dancers aim to achieve a high level of group cohesion locally known as "consonância" (consonance), as opposed to "desmantelo"(fracture, breaking up). Dancing and producing sounds in "consonância" is intended as a way to protect the group from the attacks of negative forces who may, for example, affect a dancer's body, brake an instrument, or suddenly mute the poet's whistle. A related key concept comes from the expression "fechar o Maracatu" (closing the Maracatu) that also refers to various religious and aesthetic strategies used to protect the individual and the community from perceived threats. Previous field research suggested that this expression is used as well in the acoustical and musical domain: the percussion instruments play in the fastest (160–180 BPM) and loudest way possible in order to avoid any silence or hole ("furo") in the rhythmical pattern [see Bonini Baraldi, (2021)].

Fig. 1 Three "caboclo de lança" dancers of the "Leão de Ouro de Condado" Maracatu group in Lisbon, Portugal. *Photo credit* Filippo Bonini Baraldi, 2019

The Portuguese nationally-funded research project HELP-MD,[1] "The Healing and Emotional Power of Music and Dance" explores the hypothesis that music is widely associated with healing practices in many societies around the world. This could be due to its potential to elicit and control emotions, whether this happens through symbolic associations or aesthetic meanings attributed to musical forms. Within the context of the HELP-MD project, specific focus is given to Maracatu de Baque Solto, and in December 2019, 13 members of the "Leão de Ouro de Condado" Maracatu group were invited to Lisbon, Portugal. The visit (notable as it was the first time that a Maracatu group was invited to Portugal) included several workshops, a parade through the streets of Lisbon, and a fixed location performance. In this performance, a set of recordings were obtained and form the basis of our subsequent analysis of temporal structure in this paper and the work presented in Davies et al. (2020).

The specific focus of this paper is an evaluation of the capabilities of automatic onset estimation strategies to aid in the analysis of microtiming in Maracatu. We first investigate the necessary steps to obtain clean instrument signals to annotate and then detail a semi-automatic approach for precise onset annotation in order to obtain the most accurate temporal structure data while minimising the human effort in hand-labelling. Once we obtain accurate onset annotations, we evaluate the impact of each processing step that guided the human annotator through a detailed evaluation, and in turn, how these stages affect the reliability of the the microtiming visualisation in Maracatu.

[1] https://www.help-md.eu.

The remainder of this paper is structured as follows: Sect. 2 provides an overview of the instruments of Maracatu and describes the signal acquisition process; Sect. 3 details the methodology followed to detect percussive onsets in Maracatu. In Sect. 4, we present the evaluation of our approach. In Sect. 5, we present microtiming profiles obtained under different onset detection conditions. We conclude the paper in Sect. 6 with insights on the use of computational onset detection in order to observe micro-rhythmic content in Maracatu.

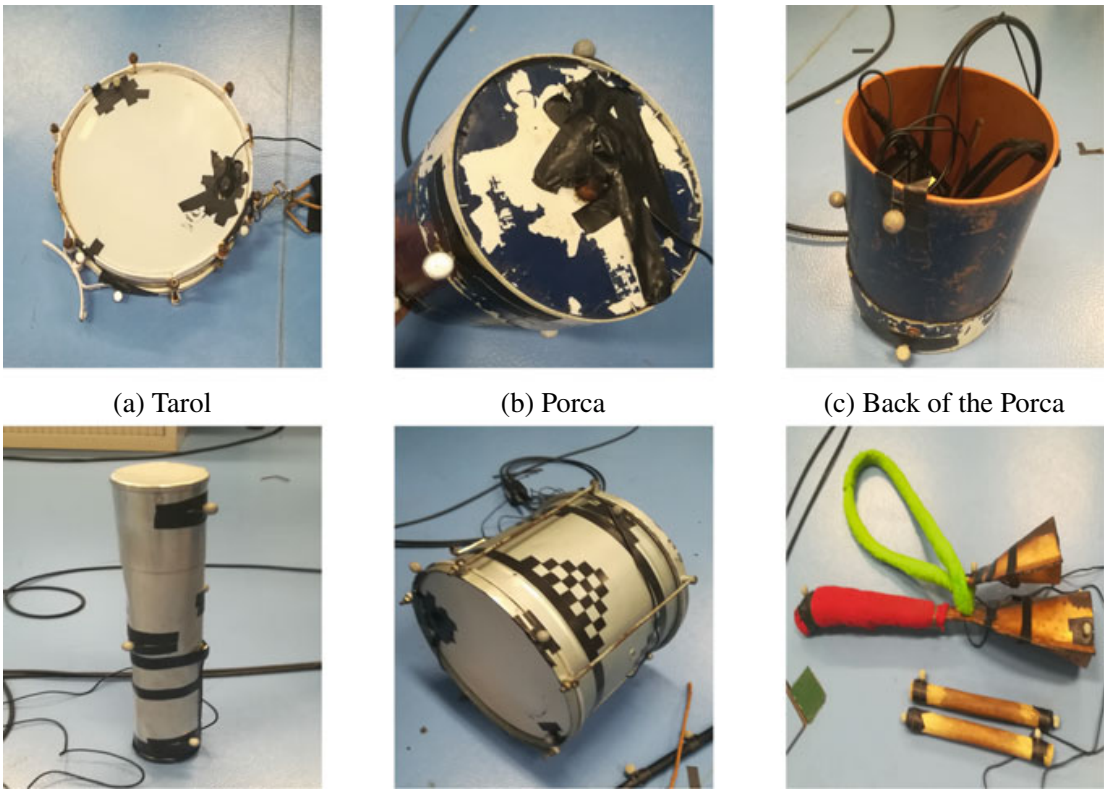
2 Maracatu Percussion Instruments and Signal Acquisition

The five percussion instruments of the terno in Maracatu are as follows: *Tarol*—a thin snare drum like instrument (Fig. 2a). *Porca*—a friction drum, also referred to as a "Cuíca" which is played with a damp cloth holding the stick (Fig. 2b and Fig. 2c. *Mineiro*—a metal tube which is filled with beads or other small objects, and which is shaken to create a rattle type sound (Fig. 2d). *Bombo*—a bass drum like instrument which is played with two sticks, one per side (Fig. 2e). We refer to the upper skin of the bombo as "Bombo High" and the lower skin as "Bombo Low." *Gonguê*—is made of iron and is comprised of two bells, with the smaller, higher pitched bell "Gonguê High" and the larger, lower pitched bell "Gonguê Low." (Fig. 2f).

Given the musical characteristics of Maracatu and the way it is performed (musicians in close physical proximity playing very loud and very fast), the process of annotating precise onset times of multiple instruments in the"terno" from mixed (e.g. stereo) recordings acquired with traditional electroacoustic microphones is extremely challenging. As such, one possibility is to look at audio source separation as a means to obtain isolated percussion signals for subsequent annotation and analysis. However, despite significant recent advances due to the use of deep neural networks Stoter et al. (2019), Hennequin et al. (2019), state-of-the-art techniques primarily target the separation of vocals, drums, and bass from popular music recordings. Given the terno is entirely comprised of percussion instruments, we should expect limited success with these approaches.

As described in detail in Davies et al. (2020), we focus instead on acquiring separated recordings at the source, via the use of contact microphones attached to each instrument of the terno. Contact microphones convert the vibrational energy of the instruments' drum surfaces into electrical energy. This recording process minimises leakage between microphone captures and consequently eases the analysis of the waveform and annotation processes. We placed two pickups on the Gonguê—one per bell, and two on the Bombo—one on each skin of the drum. For the other instruments, the Tarol, Porca, and Mineiro, we used a single pickup.

In total, we collected a dataset of isolated signals of each considered instrument for 34 pieces present in a fixed location Maracatu de Baque Solto performance, that totals approximately 22 minutes. Across 7 channels, this led to a total of 238 acquired contact microphone signals with minimum and maximum lengths of 24.9 s and 123.3 s, respectively.



(a) Tarol (b) Porca (c) Back of the Porca

(d) Mineiro (e) Bombo (f) Gonguê

Fig. 2 Maracatu de Baque Solto percussion instruments. The contact microphones are fixed with black tape and the grey balls are markers for motion capture

3 Semi-automatic Onset Detection in Maracatu

3.1 Motivation

Within the field of music information retrieval (MIR), it is common to develop computational approaches for the extraction of information from musical audio signals. In recent years, deep learning techniques have become highly prevalent and have led to significant breakthroughs in performance by leveraging the ability of deep neural networks to learn from labelled data. Of particular importance in MIR-based evaluation is the ability of these trained networks to generalise to unseen data.

In this work, the context is quite different. First and foremost, our focus is on a specific set of recordings towards the ethnomusicological analysis of the nature and role of microtiming in Maracatu. In this sense, the annotation of the onset positions can be considered a necessary intermediate step within a larger analysis pipeline, and one which must be as accurate as possible. Thanks to the largely isolated signals obtained from the contact microphones, the manual annotation of this data would be possible [e.g. using a software tool like Sonic Visualiser Cannam et al. (2006)]. However, in the graphical example in Fig. 3 which shows four contact microphone signals and their mixture, we see the dense rhythmic structure of Maracatu and hence

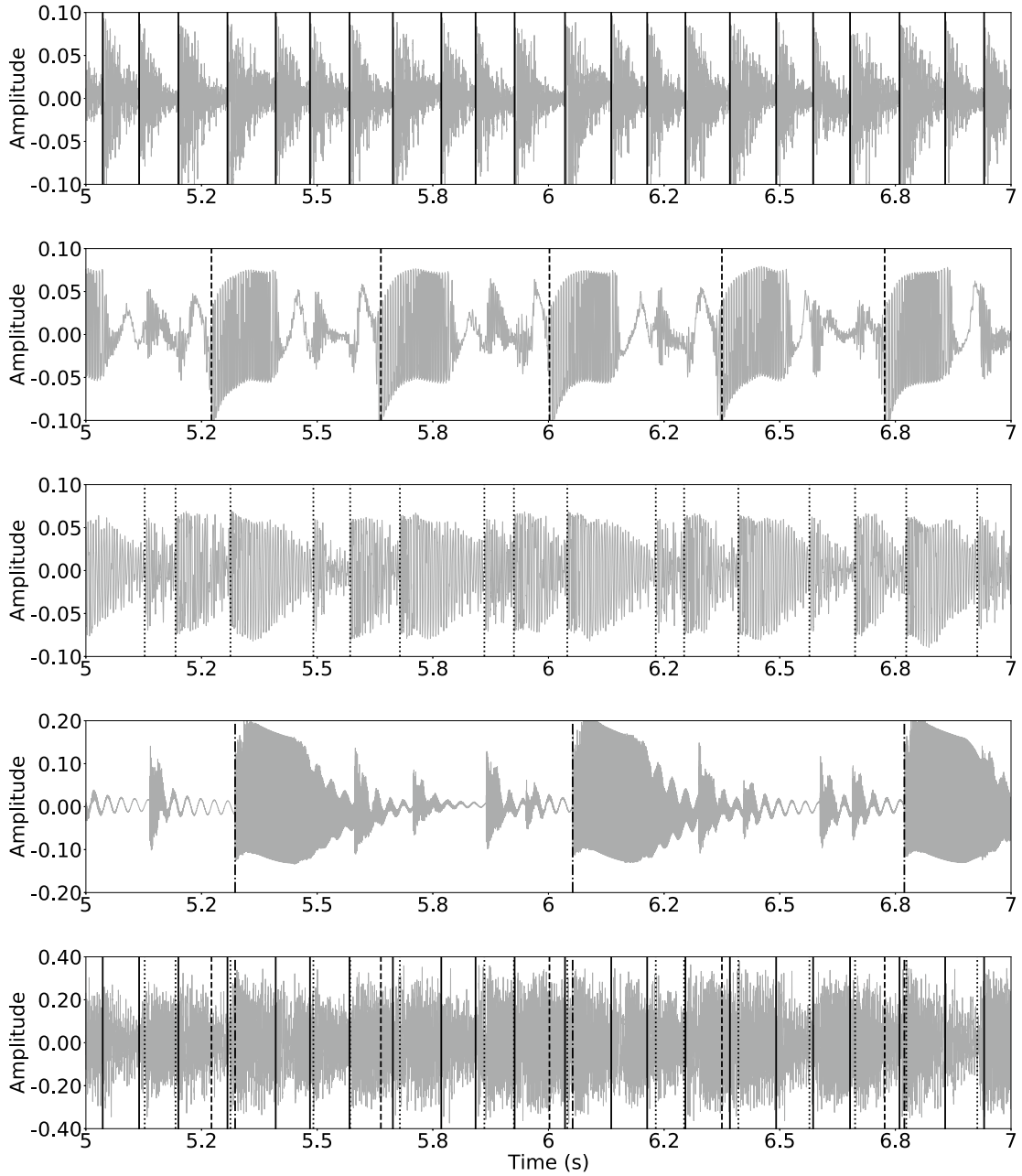


Fig. 3 Illustration of instrument signals with manually annotated onsets in approximately one bar of track #28. Top to bottom: Tarol with annotations in solid lines; Porca with annotations in dashed lines; Bombo High with annotations in dash-dotted lines; Gonguê Low with annotations in dotted lines; Audio mixture with overlaid onsets of the four instruments

the very high number of annotations necessary even across a short duration of just two seconds. Propagating this effort across 7 contact microphone signals, even for just 22 minutes of recordings from the fixed location performance, presents a considerable annotation effort.

The alternative to manual annotation is to consider the use of computational approaches for onset detection within a semi-automatic framework, where the automatic estimates are subsequently verified and corrected (as necessary) by one or more

human annotators. We pursue this semi-automatic framework for two purposes: first, as an opportunity to investigate the effectiveness of an "off the shelf" state-of-the-art onset detection method for this task; and second, to consider how context-specific information can be leveraged to guide the automatic analysis so as to minimise the amount of human interaction required for the correction of the annotations.

3.2 "Off the Shelf" Onset Detection

The existing onset detection method we use is taken from the well-known Python library, madmom Bock et al. (2016). This state-of-the-art onset detection approach [described in detail in Eyben et al. (2010)] uses a recurrent neural network (RNN) architecture and, given an input audio signal sampled at 44.1 kHz, it provides a sequence of onset times at ,a temporal resolution of 10 ms. This onset detection method has been trained on a wide variety of musical material [for details, see Bock et al. (2012)] with the purpose of being widely applicable on diverse input material. We treat this approach to onset detection as a "black box."

3.3 Maracatu-Specific Onset Detection

Since we have largely isolated signals which are percussive in nature, one might assume that the task of automatic onset detection would be relatively straightforward. However, our initial experimentation in Davies et al. (2020) with existing onset detection methods provided somewhat mixed results with noticeably poor performance on the Gonguê and Porca. Furthermore, even from a manual annotation perspective, we found the Mineiro extremely challenging to annotate in a consistent way due to the lack of well-defined attack in the waveform and thus we exclude it from further analysis in this paper.

Our hypothesis concerning the apparent low effectiveness of existing onset detection approaches hinges on two points: first, that we use contact microphone signals as input rather those obtained with traditional electroacoustic microphones, and second that the Maracatu terno is comprised of handmade instruments which may never have been observed in the training data of existing approaches. Within the context of our work, we have a fixed set of recordings, with the same performers and instruments throughout. On this basis, we investigate the use of an instrument-specific approach to onset detection in Maracatu by taking an existing deep neural network and *retraining* it specific to each individual instrument of the terno.

However, since most deep neural networks require a large amount of training data and can be slow to train (especially so of RNN approaches), we target the use of a lightweight deep neural network which can be trained very efficiently, and use "fine-tuning" to adapt the weights of an existing network rather than train from scratch. This latter aspect is particularly important, since we would like to use as little annotated

training as possible. We re-enforce that this approach is highly exploratory in nature, and embedded within a larger analysis pipeline of Maracatu, and thus our goal is not to discover an optimal means for fine-tuning but rather to investigate its potential within an musicological work flow.

Our retraining methodology is based on the temporal convolutional network (TCN) approach in Davies and Bock (2019) which was first presented for musical audio beat tracking and which we adapted for the task of onset detection. While it would have been possible to retrain a version of the RNN-based approach in madmom, the TCN offers the advantage that is particularly fast to train (even without access to specialised hardware such as GPUs) and uses proportionally far fewer weights than the recurrent neural network approach in madmom. The process involves first training the TCN on an existing onset detection dataset Bock et al. (2012). Next, for each of the instruments of the Maracatu, an instrument-adapted network is obtained by freezing all except the shallowest layers of the network (i.e. those closest to the musical surface) together with the final output layer, and fine-tuning the network on a short manually-annotated section of just 5 s per instrument. Given the lightweight nature of the TCN and the very small amount of annotated training data, this fine-tuning process could be achieved in less than a minute on a standard laptop computer, and without any need for specialised GPU hardware.

In broad terms, our goal is to build upon what the network already knows about onset detection from the initial generic training data, and then essentially to re-calibrate it so that it will focus specifically on each of the instruments of the terno. As with the madmom approach, the output of the TCN is a so-called "onset activation function" with a temporal resolution of 10 ms. To extract the sequence of detected onsets we then use the peak picking algorithm within madmom under its default parameterisation.

3.4 Precise Temporal Localisation

Most existing methods for onset detection cannot reliably identify the onset position at the sample level of the waveform because the analysis is conducted over windows of approximately 10 ms in duration Eyben et al. (2010). Indeed, in mixed recordings with multiple simultaneous instruments, there is an intrinsic uncertainty about the precise temporal location of note onsets. However, for our specific case of "clean" contact microphone signals on percussive instruments, we retain the ability to look in the waveform to attempt to identify a precise (or at least consistent way) to mark the onset locations.

We develop a straightforward approach for fine temporal localisation based on a pair of small non-overlapping sliding windows which increment at the audio sample level. Using these sliding windows around a small region surrounding an initially detected onset, we locate the point in the audio signal which leads to the maximum positive change in energy and spectral difference. In this way, we can overcome

the 10 ms quantisation effect of automatic onset detection methods and provide the means for very accurate timing analysis necessary for the estimation of microtiming.

A graphical overview of our is illustrated in Fig. 4, which we use as the means to describe our approach in detail. In the top plot we see a 1 s excerpt of the Gonguê low contact microphone signal from the same recording used in Fig. 3 with a detected onset from the Gonguê-specific TCN marked at 44.5 s as a solid vertical line—with all other detected onsets omitted for visual clarity. Close inspection of the waveform shows that the precise onset location, i.e. the point at which the energy sharply increases, does not coincide with this marker due to the 10 ms quantisation effect. In order to obtain a more accurate onset position we create a window of ± 512 samples (i.e. 1024 total samples), shown as the box surrounding the solid vertical line. Moving to the middle plot of the figure, we now zoom into the waveform over this 1024 sample range. Within this analysis window, we create two smaller windows, shown in light and dark grey, each of size 128 samples. These two windows sit adjacent to one another and slide through the 1024 sample window, one audio sample at a time. For each sample increment we measure the change in signal energy between both windows as well as the spectral flux and take the sum of these two values. The output of this measurement is shown in the lowest plot of the figure. We then find the index of the maximum value of this signal, shown as a vertical dashed line, as our more precise onset detection. In this specific case, the more precise onset is approximately 5 ms earlier than the quantised onset from the TCN. In addition to giving high temporal precision, we believe that our approach also offers the means for a highly consistent labelling of onset positions.

3.5 Semi-automatic Onset Annotation Pipeline

Given each of these components, we summarise our semi-automatic annotation pipeline as follows:

- We train the TCN using generic onset annotated data.
- Per instrument of the terno we perform fine-tuning of the TCN with a 5 s annotated region.
- On each of the recordings in our dataset we use the instrument-adaptive TCN to predict an onset activation function.
- On this signal we use the default peak picking function from the madmom library to obtain a sequence of detected onsets.
- We then perform fine temporal localisation on the detected onsets of each instrumented-adapted network.
- Finally, we load the detected onsets into Sonic Visualiser together with the corresponding waveform and perform final corrections via insertions, deletions and temporal shifts.

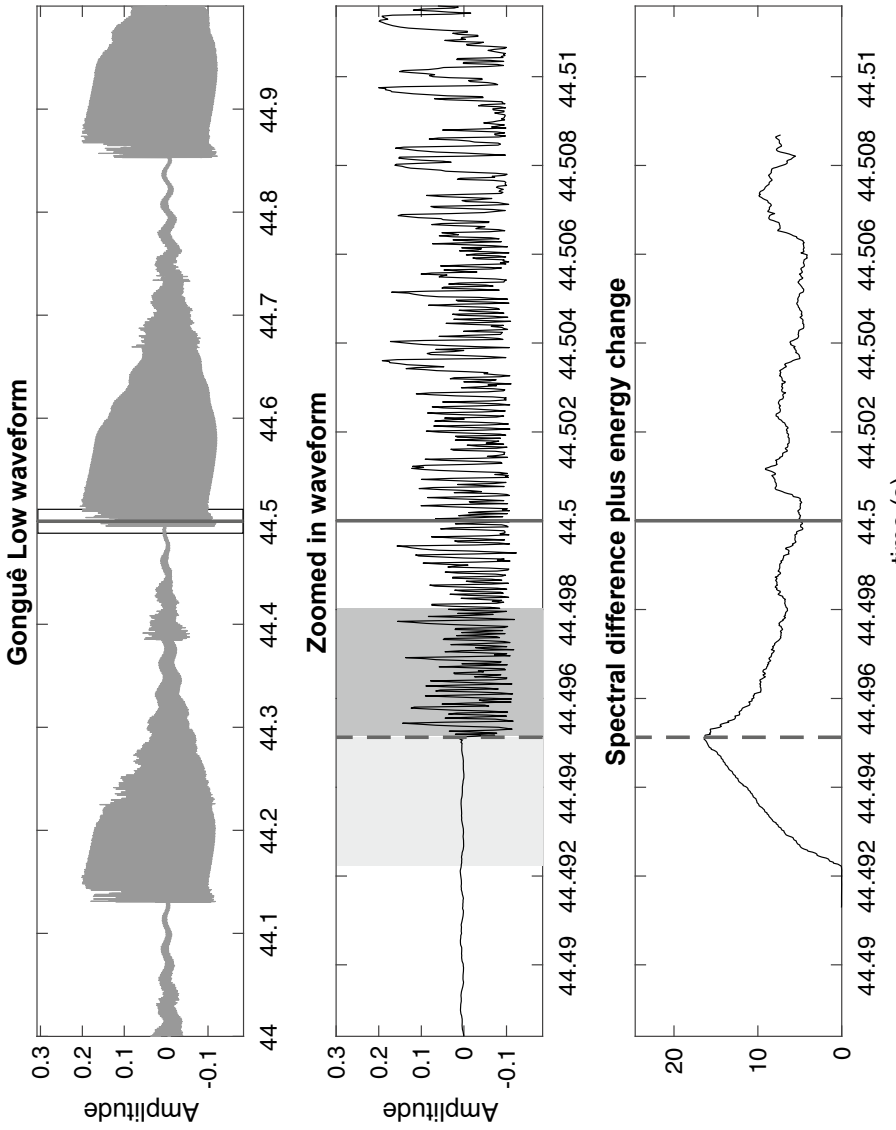


Fig. 4 Fine temporal localisation. Top plot: a 1 s excerpt of Gonguê Low with an automatically detected onset marked as a vertical line. Middle plot: a zoomed in region used to discover a more precise onset location. The light and dark grey shaded regions represent the sliding window that move through the waveform at audio sample increments. Bottom plot: the output of the energy change and spectral difference, whose maximum is marked as a dashed vertical line and used as the precise onset location

4 Evaluation of Automatic Onset Detection Methodology in Maracatu

In our evaluation we adopt the widely-used F-measure from the *mir_eval* library Raffel et al. (2014) to estimate the performance of the onset detection approaches. Given ground-truth onset data, if an estimation falls within a tolerance window around it, is considered a correct estimation (true positive). Any missed detections are considered false negatives, and detections that fall outside the tolerance windows are considered false positives.

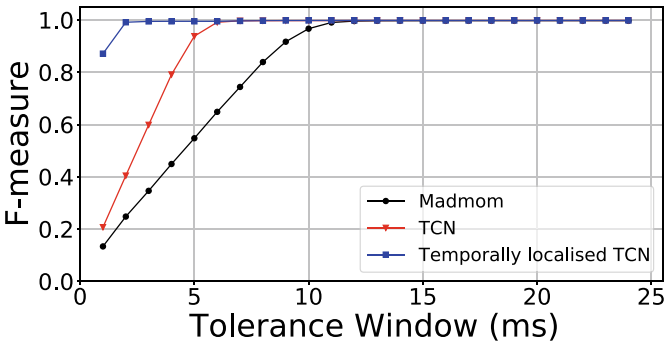
While much existing work in onset detection Collins (2005, Eyben et al. (2010), Bock et al. (2012) uses a tolerance window ±25 ms our specific interest is in the analysis of microtiming variations which may involve temporal variation at a much finer time-scale. On this basis, we perform a more extensive evaluation by calculating the F-measure repeatedly over a range of tolerance windows from ±1 ms up to ±25 ms in 1 ms increments.

Figure 5 shows the mean F-measure scores in function of increasingly wider tolerance windows per analysed instrument: (a) Tarol, (b) Porca, (c) Bombo High, and (d) Gonguê Low for 33 of the 34 pieces present in our dataset—we omit the piece used to provide the manual annotations for fine-tuning. In the plot we present the mean F-measure under three conditions: madmom (black line with dots), the instrument-specific TCN (red line with triangles), and the instrument-specific TCN after the fine temporal localisation has been applied (blue line with squares).

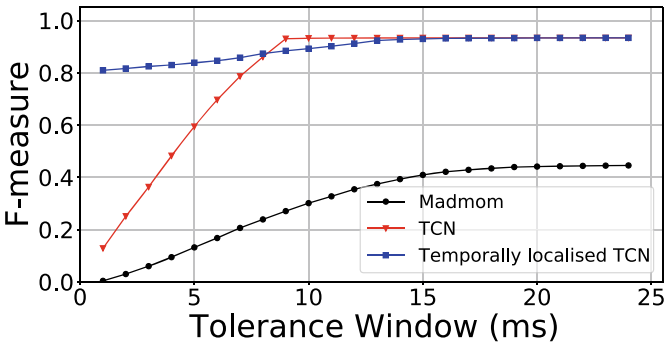
Inspection of the figure per instrument allows us to draw several conclusions. As expected, the performance across all conditions increases as the tolerance window becomes wider. Indeed, all approaches appear to saturate by ±20 ms with no further gains possible. Comparing the pattern across instruments, we see that for the Tarol and Bombo High, the performance is essentially the same under all approaches for tolerance windows greater than ±15 ms, however for Porca and Gonguê Low, we can observe a clear difference in performance, with the instrument-specific approaches obtaining much higher F-measure scores.

This result may be explained by the more conventional nature of the Tarol and Bombo High and their close similarity to existing Western drums, as opposed to the Porca and Gonguê which are timbrally quite different to the material used to train the madmom system. Inspection of the waveforms in Fig. 3 in particular for the Porca demonstrate their unusual shape. As such we can mostly clearly observe the benefit of our fine-tuning approach for these instruments. For visual clarity we do not include the performance of the baseline trained TCN (i.e. without fine-tuning), but we found its performance to be extremely similar to the madmom system.

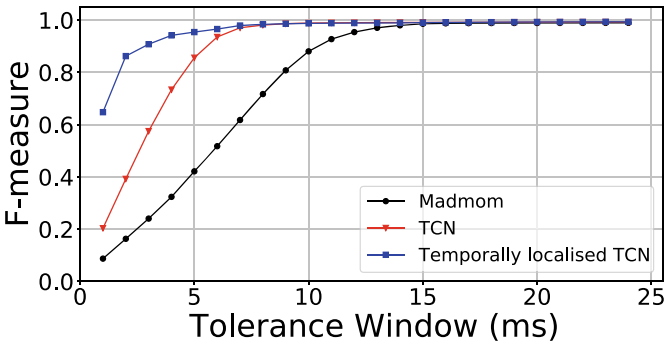
Turning our attention to temporal localisation, we see that even for for the Tarol and Bombo High, the instrument-specific TCN approaches achieve higher performance than madmom for smaller tolerance windows. This effect is even more prominent after the fine temporal localisation has been applied. Here we can observe very high F-measure scores across all instruments even for extremely small tolerance windows of just a few milliseconds.



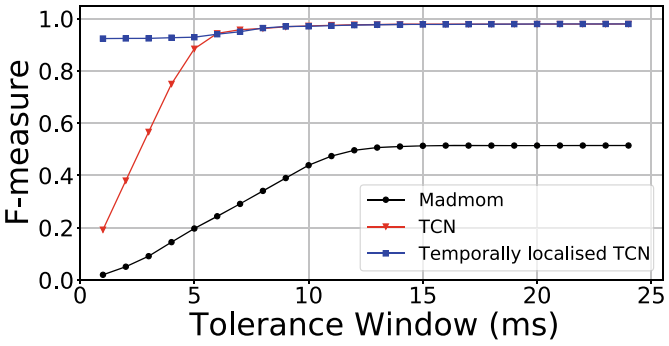
(a) Tarol



(b) Porca



(c) Bombo High



(d) Gonguê Low

Fig. 5 Mean F-measure scores in function of the tolerance window of the three automatic onset detection scenarios

We reiterate that due to the exploratory nature of the work, we do not focus too closely on the raw accuracy scores themselves, which may be improved by better optimisation when fine-tuning the networks per instrument. Instead, we highlight the trends which are present: the general improvement in accuracy via the use of instrument-specific networks as well the benefit of fine temporal localisation in providing a precise output.

5 Microtiming Visualisation in Maracatu

Having evaluting the accuracy of the different variants of our computational approach to onset detection, we now turn our attention to observing the microtiming structures present on the recordings of our Maracatu dataset. By contrasting the visualisations using the final manually verified annotations with the output of the computational approaches, our aim is to discover the impact that each processing step has on microtiming visualisation.

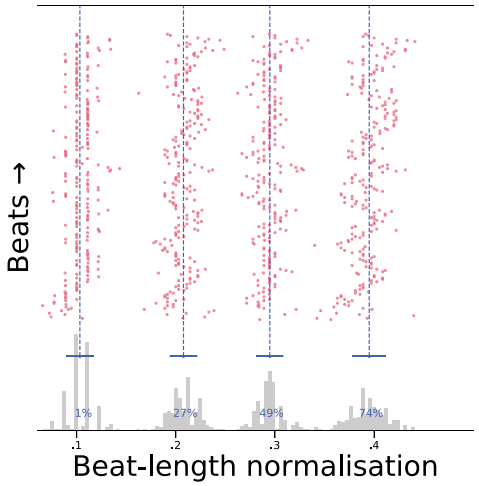
To visualise the microtiming profiles, we use the CARAT library Rocamora et al. (2019) to plot the evolution of the onset positions over the beats of a Maracatu piece and investigate the presence and nature of microtiming patterns over time.

To undertake an evaluation of the microtiming structure present in recordings, we need to obtain temporal markers that indicate both the note onset positions and metrical structure. Following previous works in microtiming analysis Naveda et al. (2011), Fuentes (2019), Jure and Rocamora (2016), we create a reference beat grid to be able to compare the locations of performed onsets.

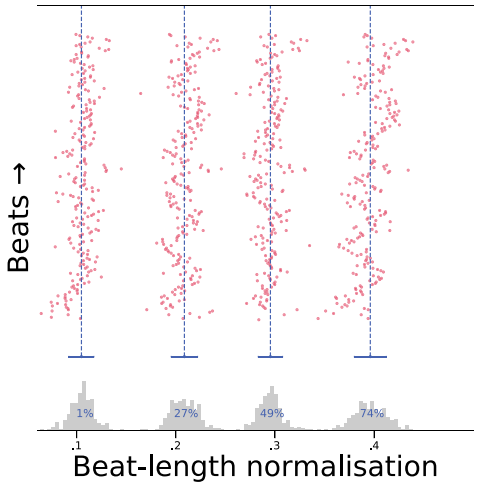
Due to the tempo variations present in the recordings, it is not possible to analyse timing data in absolute duration. For this reason, each beat interval is assigned a normalised duration of 100%, and thus a rhythmic pattern containing four sixteenth notes with equal inter-onset intervals would occur at normalised positions of 0, 25, 50, and 75%. The onsets are converted to their relative position with regard to the beats and are assigned a position relative to an isochronous metrical grid (equally distributed subdivisions within the beat) Jure and Rocamora (2016).

Although previous approaches studied within-instrument rhythmic patterns, this work, aligned with Davies et al. (2020), focuses on identifying the between-instrument microtiming deviations. We use the Gonguê Low onsets as the beat reference [since it is thought to convey the beat in Maracatu Oliveira et al. (2009)] and focus on the microtiming present in the Tarol dependent on these beats.

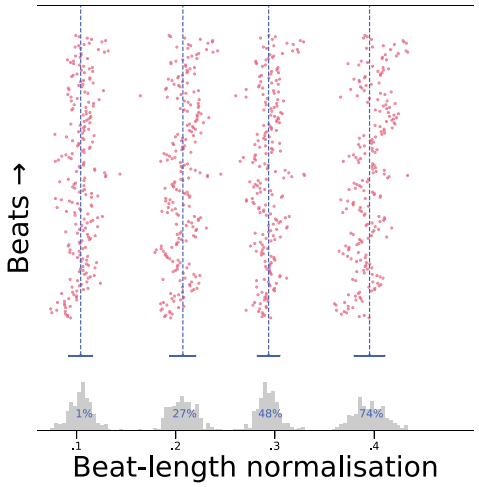
In Fig. 6 we show the microtiming profiles for one Tarol recording under three conditions: using onsets computed with the Tarol-specific TCN; these onsets after fine temporal localisation; and then compared the final output after manual correction. In the figure, the red dots represent the location of the onsets within each beat, represented horizontally through the beat-length normalisation, where the .1 tick corresponds to the first sub-beat position (that coincides with the beat) and the remaining .2, .3 and .4 ticks correspond to the second, third and fourth sub-beat positions. The beat evolution across time is displayed vertically, with time increasing from bottom



(a) Tarol-specific TCN



(b) Temporally localised TCN



(c) Manual annotations

Fig. 6 Microtiming profiles in track #28 for Tarol, with Gonguê Low estimations as the beat reference. Top to bottom microtiming profiles computed with TCN detections, temporally localised TCN detections and manually corrected annotations, respectively

to top. At the bottom of each sub-figure is a histogram of the onset locations providing a measure of the mean location of events within a group (extended by the light-blue dashed lines) and their amount of dispersion. We can use this distribution as a means to look for consistent deviations from quantised positions, while inspecting the onset positions themselves as a means to discover localised behaviour.

Looking first at the global behaviour we see that under each condition the resulting mean positions of each sub-division of the beat in the Tarol are extremely consistent at [1%, 27%, 49%, 74%] for the two TCN approaches, and then [1%, 27%, 48%, 74%] for the final manually-veritied annotations. For this specific piece, we can therefore assert that some consistent deviation from quantised positions is occuring, with the second sixteenth note slightly delayed with the third and fourth slightly anticipated.

The clearest discrepancy can be observed in the sparse and spiky nature of the distribution for the raw TCN output in the top plot for the Tarol onsets which coincide (in a notational sense) with the onsets (and hence beats) of the Gonguê Low. Here we see first-hand the quantisation effect which arises by limiting the temporal resolution to 10 ms. If we look at the temporal evolution of the onsets, we see the same pattern reflected a vertical banding for the first sixteenth note of each beat. To a lesser and lesser extent we can see the same kind of patterning in the other onsets of the Tarol, but we believe this effect is smoothed out due to the normalisation of each beat duration.

Turning to the middle plot, where we incorporate fine temporal localisation, we see that the ability to more precisely determine the onset locations removes this vertical banding behaviour entirely and the temporal evolution looks extremely similar to bottom plot which represents the output after human intervention to correct the onsets. Following initial results observed in Davies et al. (2020) we can see some evidence of a dynamic use of microtiming, where the trend of the red dots appears to move ahead and behind the beats marked by the Gonguê Low. Inspection of the plots shows this trend to be most pronounced for the second and fourth sixteenth notes.

While we can begin to draw insight about the use of microtiming in Maracatu, our goal in this work is rather to demonstrate the similarity in the visual representation contrasting automatic onset detection (with fine temporal localisation) and how this compares to the final output after manual correction. In this sense, we seek to highlight the benefit of contextually-guided computational analysis as a means to reduce the work load placed on a human annotator.

6 Conclusions

In this work we conduct a rigorous analysis of computational onset detection methods when compared to manual annotations of Maracatu recordings. We contrast a "black box" state-of-the-art approach from the well-known madmom library with instrument-specific methods using fine-tuning and fine temporal localisation. In developing these approaches our goal is not to understand the generalisation properties of the automatic approaches—as would be customary in the MIR onset detection

literature, but rather to see the cumulative impact of each processing step in arriving at a highly accurate output which can be used for subsequent ethnomusicological analysis. Within our evaluation we noted that the use of an instrument-specific approach was most beneficial for the Porca and Gonguê Low recordings, which we believe had a profound impact in reducing the amount of human interaction required to produce a verified set of onset annotations..

In this kind of musicological work, which ultimately focuses on a small set of recordings from single set of performers on fixed instruments, we cannot hope to obtain "big data" in the sense which the term is applied in other domains using DNNs. Thus, from a technical perspective, this work raises important questions about how we can develop techniques that can learn from "small data", or adapt existing trained models so they can be effective in highly constrained circumstances. In this context we believe that fine-tuning is a promising step towards creating application-specific and hence more accurate analysis for end-users. In a broad sense, we hope this work can help advance the ongoing discussion over the use of computational methods to aid musicologists. In future work we look to extend our contribution by creating streamlined tools with straightforward workflows which can enable the user-specific adaption of deep neural networks within data-constrained and under-explored problems.

Acknowledgements This work is supported by Portuguese National Funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project "The Healing and Emotional Power of Music and Dance" (HELP-MD), PTDC/ART-PER/29641/2017.

This work is funded by national funds through the FCT—Foundation for Science and Technology, I.P., within the scope of the project CISUC-UID/CEC/00326/2020 and by European Social Fund, through the Regional Operational Program Centro 2020 as well as by Portuguese National Funds through the FCT—Foundation for Science and Technology, I.P., under the project IF/01566/2015.

Magdalena Fuentes is a faculty fellow in the NYU Provost's Postdoctoral Fellowship Program at the NYU Center for Urban Science and Progress and Music and Audio Research Laboratory.

References

Böck, S., Arzt, A., Krebs, F., & Schedl, M. (2012a). Online real-time onset detection with recurrent neural networks. In *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, York, United Kingdom.

Böck, S., Krebs, F., & Schedl, M. (2012b). Evaluating the online capabilities of onset detection methods. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, (pp. 49–54).

Böck, S., Korzeniowski, F., Schlüter, J., Krebs, F., & Widmer, G. (2016). Madmom: A new python audio and music signal processing library. In *24th ACM International Conference on Multimedia*, Amsterdam, The Netherlands, (pp. 1174–1178).

Bonini Baraldi, F. (In Press). Inveja e corpo fechado no Maracatu de baque solto pernambucano. Sociologia & Antropologia. (In Portuguese).

Cannam, C., Landone, C., Sandler, M. B., & Bello, J. P. (2006). The sonic visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of 7th International Conference on Music Information Retrieval*, Victoria, Canada, (pp. 324–327).

Collins, N. (2005). A comparison of sound onset detection algorithms with emphasis on psycho acoustically motivated detection functions. In *Proceedings of 118th Audio Engineering Society Convention*, Barcelona, Spain, (pp. 6363).

Davies, M. E. P., & Böck, S. (2019). Temporal convolutional networks for musical audio beat tracking. In *Proceedings of the 27th European Signal Processing Conference*, A Coruña, Spain. https://doi.org/10.23919/EUSIPCO.2019.8902578.

Davies, M. E. P., Fuentes, M., Fonseca, J., Aly, L., Jerónimo, M., & Baraldi, F. B. (2020). Moving in time: Computational analysis of microtiming in maracatu de baque solto. In *Proceedings of 21st International Society for Music Information Retrieval Conference*, Montreal, Canada, (pp. 795–802).

Eyben, F., Böck, S., Schuller, B., & Graves, A. (2010). Universal onset detection with bidirectional long-short term memory neural networks. In *Proceedings 11th International Society for Music Information Retrieval Conference*, Utrecht, The Netherlands, (pp. 589–594).

Fuentes, M. (2019). Multi-scale computational rhythm analysis: A framework for sections, downbeats, beats, and microtiming. Ph.D. Thesis, Université Paris-Saclay. NNT : 2019SACLS404.

Hennequin, R., Khlif, A., Voituret, F., & Moussalam, M. (2019). Spleeter: A fast and state-of-the art music source separation tool with pre-trained models. In *Late Breaking/Demo at the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

Jure, L., & Rocamora, M. (2016). Microtiming in the rhythmic structure of candombe drumming patterns. In *Fourth International Conference on Analytical Approaches to World Music*, New York, United States, (pp. 8–11).

Koen, B., Lloyd, J., Barz, G., & Brummel-Smith, K. (2008). *The Oxford handbook of medical ethnomusicology*. New York: Oxford University Press.

Naveda, L., Gouyon, F., Guedes, C., & Leman, M. (2011). Microtiming patterns and interactions with musical properties in samba music. *Journal of New Music Research*, *40*(3), 225–238. https://doi.org/10.1080/09298215.2011.603833.

de Oliveira, S. C., Resende, T. S., & Keays, P. M. (2009). *Batuque Book: Maracatu Baque Virado e Baque Solto*. Recife: Edição do autor.

Raffel, C., McFee, B., Humphrey, E. J., Salamon, J., Nieto, O., Liang, D., & Ellis, D. P. W. (2014). mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Taipei, Taiwan, (pp. 367–372).

Rocamora, M., Jure, L., Fuentes, M., Maia, L., & Biscainho, L. (2019). Carat: computer-aided rhythmic analysis toolbox. In *Late Breaking/Demo at the 20th International Society for Music Information Retrieval Conference*, Delft, The Netherlands.

Stöter, F. R., Uhlich, S., Liutkus, A., & Mitsufuji, Y. (2019). Open-unmix—a reference implementation for music source separation. *Journal of Open Source Software, 4*, 1667. https://doi.org/10.21105/joss.01667.