MIR WEB INTERFACE FOR SHAPING MUSICAL CREATIVITY

Tom Collins
Music Artificial Intelligence Algorithms, Inc.
contact@musicintelligence.co

Peter Knees
Department of Computational Perception
Johannes Kepler University Linz
peter.knees@jku.at

Christian Coulon
Music Artificial Intelligence Algorithms, Inc.
contact@musicintelligence.co

ABSTRACT

We designed a Web-based, piano-roll interface using the Web Audio API and associated JavaScript (JS) packages, and embedded a Music Information Retrieval (MIR) technology for automatic note/chord suggestion that extends what a user has already written. [1] In a preliminary investigation, composers used the interface to devise two loops in the style of electronic dance music (EDM) – one loop with and one loop without the "suggest" button. Users reported enjoying the suggestion functionality (mean rating 5.167 on a scale 1–7, *sd* = 1.528), and time-lapse composition edits revealed that suggestion requests accounted for approximately 15% of user actions. Investigations are underway as to whether the suggestion functionality enhances the perceived creativity of user compositions, in a study where EDM listeners rate finished loops. Attendees of the demo session will be able to listen to composed loops and try out the interface for themselves.

1. INTRODUCTION

Over recent years, there has been an increasing interest in how MIR technology might be put to creative use [3, 6, 8, 10, 12]. So-called *creative MIR* [8] involves an MIR algorithm helping a user to create some musical entity. Even outside of creative MIR, relatively little is known about how users compose with conventional software (e.g., Logic Pro, Sibelius). One study [2] tracked the behavior of an individual composer over a period of three years, including autosave MIDI files. Research on groups of composers has investigated requirements and behaviors as a function of musical expertise, age, and task [5, 7]. Web browsers, with their ability to capture user actions, are well placed to investigate the details of how users compose, and the recent emergence of the Web Audio API [1] and related packages [9, 11] present an ideal opportunity for embedding MIR technology into easily distributable interfaces.

[1] At http://musicintelligence.co/#listen-sec there are examples of output and a link to the interface.

2. INTERFACE AND MIR SUGGESTION TECHNOLOGY

The clickable piano-roll interface depicted in Fig. 1 builds on JS package called NexusUI [11]. Pitch is indicated on the y-axis using variegated rows for white and black notes on the piano keyboard, and green for pitch-class C. Measures and main beats are indicated on the x-axis using variegated columns. An accompanying drum track is heard but not seen when "Play" is pressed, and progress through the loop is indicated by orange coloring of the taller row at the bottom of the piano roll. A second JS package called Tone.js is used to link the clicking of piano-roll cells to synth pad sounds of appropriate start time, pitch, and duration, allowing realtime interactive editing of the loop and accurate synchronisation with the drum track [9]. The "Get Suggestion" button is circled by a red dashed line in Fig. 1. Next to this is an "Undo Suggestion" button and an "I Am Finished" button for submitting a completed loop. Each time a user adds, removes, or edits the duration of a note, or requests/undoes a suggestion, a combination of HTML, JS, and PHP results in this edit being time-stamped and stored for analysis.

When a user clicks "suggest", a Markov-based algorithm analyzes their in-progress composition and returns a short continuation. This Markovian approach has been used in previous research to generate melodic continuations [3], as well as to generate entire passages of music [4], but its extension to polyphonic note suggestion (what we mean by *chords*) and embedding in an easily distributable Web interface are novel. The suggested notes are generated using a model with state space S consisting of beat and relative MIDI note numbers (MNNs), and a transition matrix T detailing the probabilities of transitions between states $s \in S$. The state space and transition matrix are populated by analysing a corpus of 50 EDM excerpts. When a user clicks the "suggest" button, their in-progress composition is converted into the state-space representation. The last composed state s_q is used to query the transition matrix T. If s_q is observed in the corpus (that is, $s_q \in S$), then there will be $r > 0$ corresponding continuations $t_1, t_2, \ldots, t_r$ in T. If s_q is not observed in the corpus, then no suggested continuations can be returned to the user. When $r > 0$, one continuation is selected at random, t', and appended to the user's composition states to give $s_1, s_2, \ldots, s_q, t'$. Then t' is used to query

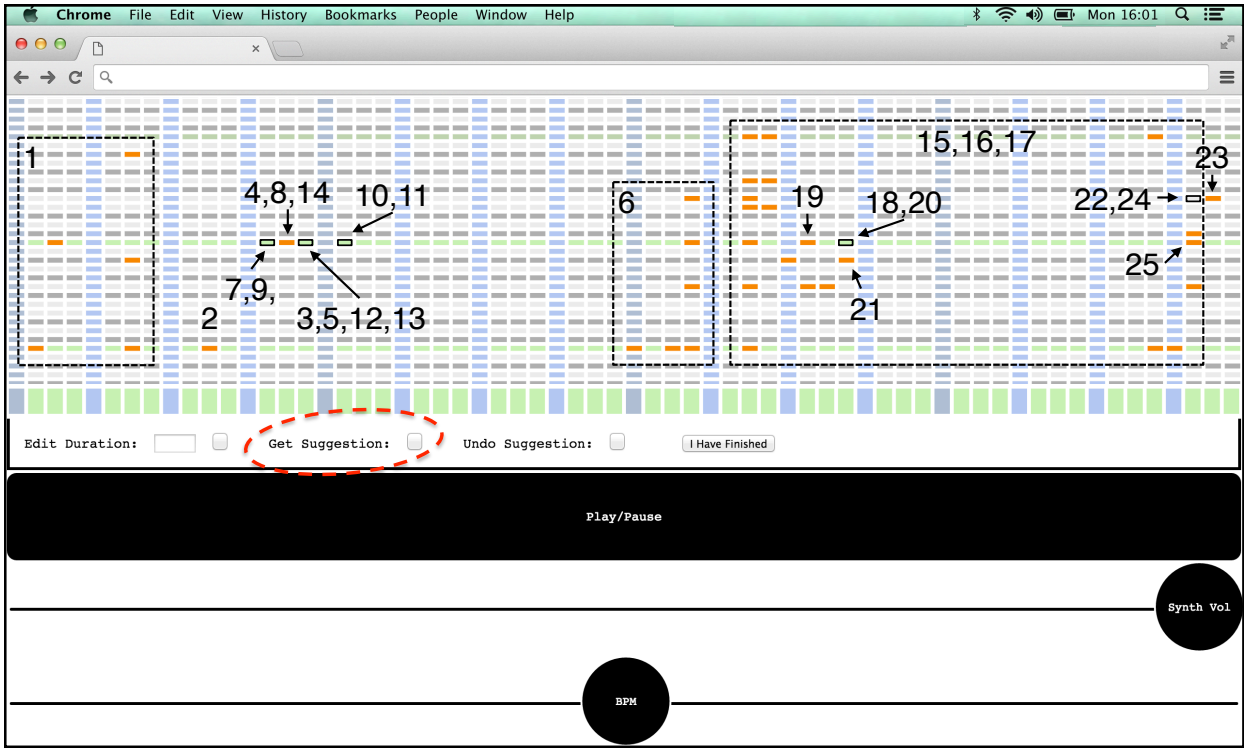

Figure 1. Browser-based, piano-roll interface, with orange oblongs of certain x- and y-values corresponding to notes with certain start times and pitch values. Numbers 1-25 indicate the time-lapse data for this composition session by participant 2, with dashed black lines bounding suggestions and black-edged oblongs indicating added-then-removed notes.

the transition matrix in the same way, and a randomly selected continuation t'' is appended. This process is repeated until five states have been appended, at which stage the states are converted from beat-relative-MNNs back into notes with absolute start times and pitch values, and the piano-roll interface is updated. If the state space is sufficiently dense, then continuations will sample several different songs, leading to musical events that "sound new".

3. USER STUDY

We conducted a study in which twelve international, professional music producers used the interface in Fig. 1 to compose two four-measure EDM-style loops accompanying a given drum track. One loop was composed in the presence of a "suggest" button and the other loop was not. The aim of the study was to investigate composers' opinions and edit behavior under suggestion-enabled versus suggestion-free conditions, and to shed light on this particular instance of using MIR to shape musical creativity.

Numbers 1–25 in Fig. 1 indicate note-by-note edits of participant 2 in the suggest condition. The user begins by requesting a suggestion (dashed black bounding box labeled 1), then makes some additions/removals following the suggested events (2–5), requests another suggestion (6), followed by further edits of intervening material (7–14), etc. [2] These suggestion requests are integrated among ordinary note edits, which is typical of other users too, and is encouraging because it suggests that users were able to assimilate the suggestion functionality into their creative processes. Post-task questionnaires and interviews indicate that participants enjoyed the suggestion function (mean rating 5.167 on a scale 1–7, $sd = 1.528$). Total edits performed by each user ranges from 10 for participant 11 to 82 for participant 3. Mean suggestion undos as a percentage of total requests is 19.25%. That is, a suggestion might be considered successful ∼80% of the time.

[2] If a user's loop is empty when (s)he clicks suggest, suggestions are returned based on a default query state of beat 1 and relative MNN 0.

4. CONCLUSIONS AND FUTURE WORK

There are myriad ways in which MIR algorithms might be embedded in interfaces so as to shape—and perhaps eventually enhance—human musical creativity. This extended abstract has described one such interface and evaluation methodology for studying a stylistically constrained instance of creative MIR. When asked if they would consider incorporating such suggestion functionality into their compositional practice, all but one interviewee responded positively: "Something I would use, definitely" (participant 8); "It's definitely got a place. . .yeah for sure" (participant 12). Future work will apply creative MIR to other types of compositional task and styles of music, and address the question of whether such technology is perceived by independent listeners as enhancing musical creativity.

5. ACKNOWLEDGMENTS

This work is supported in part by the European Union under the FP7 Framework (Giant Steps, grant no. 610591).

6. REFERENCES

[1] Paul Adenot and Chris Wilson. Web Audio API GitHub repository. https://webaudio.github.io/web-audio-api/. Retrieved August 5, 2015.

[2] David Collins. A synthesis process model of creative thinking in music composition. *Psychology of Music*, 33(2):193–216, 2005.

[3] Tom Collins and Christian Coulon. FreshJam: suggesting continuations of melodic fragments in a specific style. In *Proceedings of the International Workshop on Musical Metacreation*, Palo Alto, CA, 2012. AAAI Press.

[4] Tom Collins, Robin Laney, Alistair Willis, and Paul H. Garthwaite. Developing and evaluating computational models of musical style. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 30(1):16–43, 2016.

[5] Maria Helena De Lima, Damián Keller, Marcelo Soares Pimenta, Victor Lazzarini, and Evandro Manara Miletto. Some features of children's composing in a computer-based environment: the influence of age, task familiarity and formal instrumental music tuition. *Journal of Music, Technology and Education*, 5(2):195–222, 2012.

[6] Ángel Faraldo, Cárthach Ó Nuanáin, Daniel Gómez-Marín, Perfecto Herrera, and Sergi Jordà. Making electronic music with expert musical agents. In *Proceedings of the International Society for Music Information Retrieval Late Breaking/Demo Session*, Málaga, Spain, 2015.

[7] Allan Hewitt. Some features of children's composing in a computer-based environment: the influence of age, task familiarity and formal instrumental music tuition. *Journal of Music, Technology and Education*, 2(1):5–24, 2009.

[8] Eric J. Humphrey, Douglas Turnbull, and Tom Collins. A brief review of creative MIR. In *Proceedings of the International Society for Music Information Retrieval Late Breaking/Demo Session*, Curitiba, Brazil, 2013.

[9] Yotam Mann. Interactive music with Tone.js. In *Proceedings of the Web Audio Conference*, Paris, France, 2015.

[10] Matt McVicar, Satoru Fukayama, and Masataka Goto. AutoLeadGuitar: automatic generation of guitar solo phrases in the tablature space. In *Proceedings of the IEEE International Conference on Signal Processing*, pages 599–604, Hangzhou, China, 2014.

[11] Ben Taylor, Jesse Allison, Will Conlin, Yemin Oh, and Danny Holmes. Simplified expressive mobile development with nexusui, nexusup and nexusdrop. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 257–262, London, UK, 2014.

[12] Richard Vogl, Matthias Leimeister, Carthachó Nuanáin, Sergi Jordà, Michael Hlatky, and Peter Knees. An intelligent interface for drum pattern variation and comparative evaluation of algorithms. *Journal of the Audio Engineering Society*. In press.