AutoGuitarTab: Computer-Aided Composition of Rhythm and Lead Guitar Parts in the Tablature Space

Matt McVicar, Satoru Fukayama, and Masataka Goto

***Abstract*—We present AutoGuitarTab, a system for generating realistic guitar tablature given an input symbolic chord and key sequence. Our system consists of two modules: AutoRhythmGuitar and AutoLeadGuitar. The first of these generates rhythm guitar tablatures which outline the input chord sequence in a particular style (using Markov chains to ensure playability) and performs a structural analysis to produce a structurally consistent composition. AutoLeadGuitar generates lead guitar parts in distinct musical phrases, guiding the pitch classes towards chord tones and steering the evolution of the rhythmic and melodic intensity according to user preference. Experimentally, we uncover musician-specific trends in guitar playing style, and demonstrate our system's ability to produce playable, realistic and style-specific tablature using a combination of algorithmic, user-surveyed and expert evaluation techniques.**

***Index Terms*—Algorithmic composition, computer-aided composition (CAC), symbolic music processing.**

I. Introduction

THE generation of music using automated or semi-automated means has an extensive and rich history yet remains an active area of research [1]. The advent of affordable digital computing has facilitated a rapid increase in research into algorithmic composition systems, driven by scientific curiosity [2], analysis and replication of style [3], and the potential pedagogical benefits these systems offer [4].

A particularly interesting area of algorithmic music generation is Computer-Aided Composition (CAC), in which the compositional task is shared between computer and human. In this paper we propose that the field of CAC is sufficiently mature that investigation into more specialized models, such as those geared towards particular instruments, is now timely.

The guitar represents an interesting case study in this regard, since the design of the instrument limits (by maximum hand span) the playability of certain pieces. Furthermore, the pitch ranges of guitar strings significantly overlap, meaning that there is no unique way to play a given note. The situation for polyphonic scores naturally is significantly more complex, since each constituent note can be played in a number of different positions. This ambiguity has led to the development of tablature notation (or simply *tab*, plural *tabs*), which specifies the exact position (string and fret number[1]) to play each note (see Fig. 2).

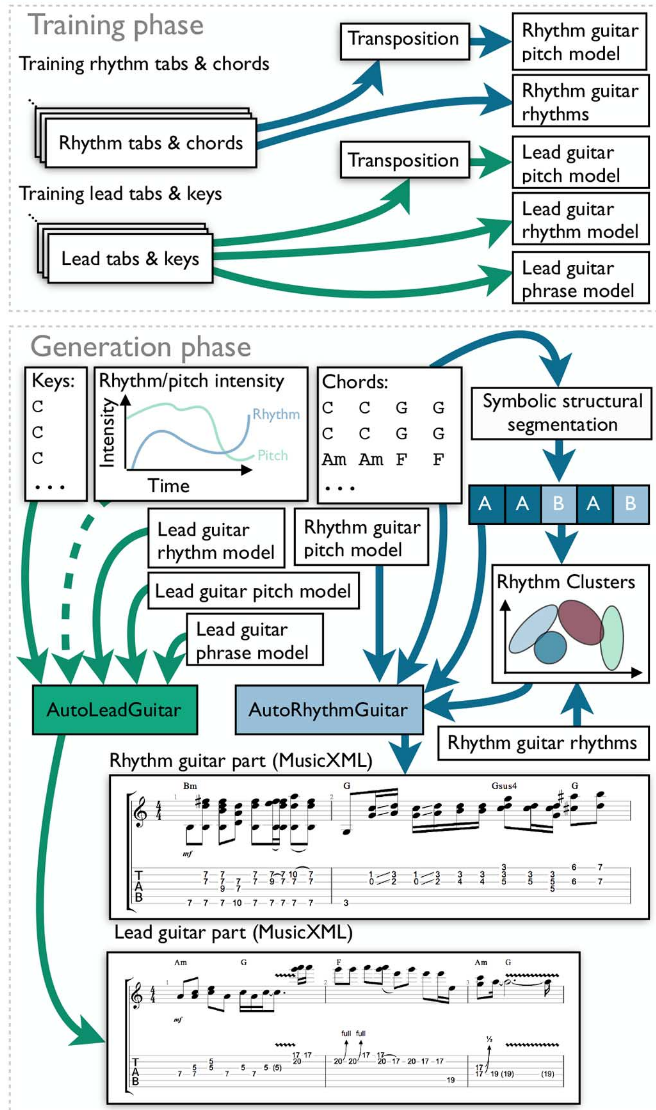


Fig. 1. High-level overview of AutoGuitarTab. AutoLeadGuitar processes are shown in green, AutoRhythmGuitar processes in blue. See Section I-C for details.

[1]frets are logarithmically-spaced ridges on the guitar neck which result in distinct semitone-spaced notes.

Manuscript received October 30, 2014; revised March 04, 2015; accepted March 21, 2015. Date of publication April 06, 2015; date of current version May 08, 2015. This work was supported by OngaCREST, CREST, JST. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Woon-Seng Gan.

The authors are with National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki, 305-8568, Japan (e-mail: mattjamesmcvicar@gmail.com; s.fukayama@aist.go.jp; m.goto@aist.go.jp).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASLP.2015.2419976

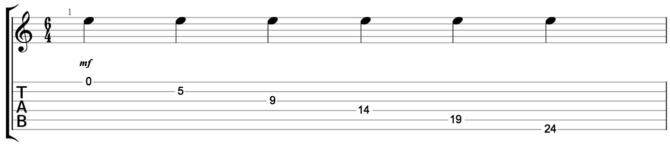


Fig. 2. Traditional notation (upper stave) and guitar tablature (lower stave) showing the troublesome one-to-many relationship between traditional notation and the guitar fretboard.

In the context of music generation, the non-bijective map from score to tab implies that compositions generated through existing algorithmic means may produce music which is extremely challenging or impossible to play on the guitar, limiting their use for teaching purposes and personal study. To overcome this problem, in this paper we introduce the composition of music directly in guitar tablature, and place probabilistic constraints on the sequences of notes which our system produces in order to maintain playability.

A. Motivation

Our primary motivation for studying guitar-specific CAC is that output from our system might be used in a compositional context, in which a composer wishes to write a guitar part for a new or existing song but lacks sufficient familiarity with the instrument to do so. Worth noting at this point is that the guitar is a versatile instrument pervasive in many forms of popular music [5], meaning the scope of this work is not particularly limited by our choice of instrument.

Further motivation for this work is more pedagogical in nature: our system could be used to teach amateur guitarists the possible creative approaches to playing rhythm or lead guitar over a given chord or key sequence (see I-B). Finally, from an academic perspective we are interested to see to what extent guitar compositions are player-specific, and if guitarists can be identified by their playing style. Analysis of style and form therefore constitutes an important final impetus of the current study.

B. Challenges and Proposed Solutions

There are several challenges which must be overcome in the development of an automatic composition system. Some of these are fairly universal in nature, whilst others more specific to algorithmic music creation for the guitar. We outline some of these challenges and our proposed solutions below.

Markov-chain-based approaches to algorithmic composition [6] (on which the majority of our approach is based) are popular within the field of CAC owing to their low complexity and highly intuitive nature. However, it is well-known that these techniques fail to model long-term behavior beyond the degree of the model (see II-A). We counter this in our work by embedding structural information in our rhythm guitar compositions, composing guitar solos in distinct musical phrases, and allowing users of our system to specify *intensity curves* outlining the desired evolution of the rhythmic and melodic energy of the composition.

Capturing the notion of style in guitar playing is another challenge we face in this work. The layout of the instrument (in particular the position options for each note) we believe plays an important role in this regard. For example, a specific chord can be outlined in a large number of ways by a musician using different combinations of rhythms and chord tones. However, on the guitar there are also different ways of fingering basic chord shapes with the same pitches, allowing practitioners of the instrument to explore different hand positions and chord shapes for a set chord.

Indeed, certain hand positions for chords are beneficial for playing arpeggios, adding additional melodic notes, or simply for the different timbre they produce [7]. The same can be said for lead guitar playing, with certain scale shapes facilitating the use of particular guitar-specific ornaments such as string bends, legato phrases or slides. For examples of this aspect of style in existing pieces, see Fig. 3.

We therefore consider position choice, along with basic rhythm and pitch selection, to be an element of a guitarist's style (both for rhythm and lead playing). We also consider use of muted notes (percussive sounds produced by relaxing the fretting hand and strumming across the strings), string bends (notes which are plucked and subsequently pulled away from their original position resulting in a smooth increase in pitch due to an increase in string tension), hammer-ons (notes which are forcefully brought onto the fretboard at frets higher than existing notes on the same strings), pull-offs (analogous to hammer-ons), and slides (glissando across frets) to constitute part of what makes a player unique. Note that existing algorithmic compositional models which generate music in traditional notation are unable to exploit these attractive aspects of the guitar, since knowledge of fingering positions is necessary for determining the applicability of these techniques.

C. System Overview

A graphical outline of our system is presented in Fig. 1. In the training phase, we collect existing digital tablatures annotated with chords and keys, from which pitch-invariant models for rhythm and lead guitar are generated via transposition. Rhythm guitar measures are stored for further processing; lead guitar models for phrasing and rhythm are trained from data.

In generation, we begin by conducting an automatic structural segmentation of the input chord sequence. This segmentation is used to cluster rhythm guitar rhythms into distinct groups. These rhythm clusters, along with the input chord sequence, pitch model for rhythm guitar, and the structural segmentation are then fed into the AutoRhythmGuitar module. A digital tablature in MusicXML format is written as output.

For lead guitar, a key sequence input is required. Optional intensity contours for rhythm and pitch are used in conjunction with a rhythm, pitch and phrasing model to produce a lead guitar digital tablature.

MusicXML is used as the output format in this work owing to its flexibility (guitar finger positions as well as bends and other guitar-specific ornaments can be written and read in MusicXML) and portability (files can easily be imported into many existing software packages for visualization and audio synthesis).

D. Contributions and Paper Structure

This paper is an extension of two of our previous publications on automatic generation of rhythm [8] and lead [9] guitar parts. The main contributions of the current work are: a new algorithm

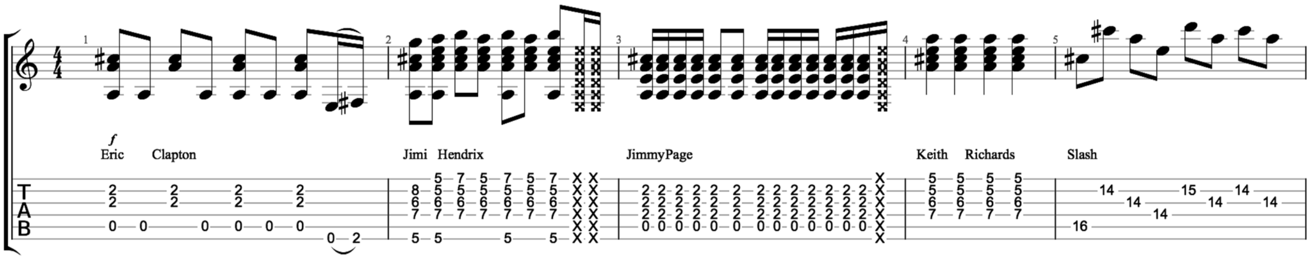


Fig. 3. The approaches five popular rhythm guitarists have taken to playing over an A major chord. Guitarists and excerpts, from left to right: Eric Clapton, "Wonderful Tonight" (Clapton); Jimi Hendrix, "Stone Free" (Hendrix); Jimmy Page, "Immigrant Song" (Page/Plant); Keith Richards, "The Last Time" (Jagger/Richards); Slash, "Sweet Child O' Mine" (Rose/Slash/Stradlin/McKagen/Adler). Notation: slurs indicate hammer-ons/pull-offs, 'X' indicates muted notes.

for the detection of segment boundaries based on genetic algorithms, the introduction of polyphony and intensity curves in AutoLeadGuitar, and a thorough evaluation of our system using algorithmic tests, user studies and expert consultation.

The remainder of this paper is organized as follows. In Section II, we discuss the existing literature relevant to the current study. Section III describes our structural segmentation algorithm. An overview of the two main modules of our system, AutoRhythmGuitar and AutoLeadGuitar, are outlined in Section IV and V. Our algorithms are evaluated in Section VI, before we conclude in Section VII.

II. Relevant Work

A. Computer-Aided Composition

The field of computer-aided composition is a broad and active research topic, making a complete literature review of the subject beyond the scope of this paper (an excellent overview can be found in [1]). As such, the current subsection focuses on the literature most relevant to approach we take in this work: Markov-chain-based generation of harmony.

Markov initially investigated sequences of time dependent variables in the context of text analysis in 1913 [1], but it was not until the mid-twentieth century that Markov chains' application to algorithmic composition was considered by Olson [10]. In this investigative work, Olson built first and second-order Markov models of pitch and rhythm from a set of eleven existing pieces, transposing all songs to a common key. The first system to use Markov chains for composition was developed by Hiller and Isaacson [2], whose *Illiac Suite* used transition probabilities to assign intervals for each of the constituent instruments. In slightly later work with Baker [11], transition probabilities for pitch, note duration, dynamics, number of non-rest notes, and playing style were learnt from an existing work. Three works by Xenakis in 1959 [12] also make use of Markovian analysis, used to control systems of sinusoidal sounds, violins, and an entire string orchestra.

In the subsequent decades, Markov chains were explored by many researchers, including Zaripov and Russell [13], Conklin and Witten [14], and Ponsford *et al.* [15]–an excellent overview of the use of Markov chains in the latter half of the twentieth century can also be found in [16]. More recently, Pachet [17] has been investigating the use of Markov chains for generation of musical melodies, leading to the development of a number of systems for use by both trained [3] and untrained [18] musicians. A common criticism of Markovian models is that they fail to model long-term behavior beyond the order of the model [19] and can lead to plagiarism by simply replicating high-probability state sequences, especially with high order models. Pachet and his collaborators acknowledge and counter this by placing constraints on the maximum permissible order of sequence which are repeated verbatim [20].

B. Algorithmic Fingering and Arrangement

Following the terminology from Hori *et al.* [21], we describe the process of producing the most comfortably playable tablature from a score as computing a guitar *fingering*. To do so for complex pieces is non-trivial even for skilled guitarists, and may not even be possible. In these cases, minimally altering a piece to ensure it is playable on the guitar is described as determining an *arrangement* of the piece.

Manually producing fingerings and arrangements are sufficiently irksome tasks that some researchers have investigated methods of automating these procedures. Sayegh [22] introduced an *optimal path paradigm* to minimize fingering movement, extended by Radicioni *et al.* [23] to optimize phrase-level, rather than global, fingering movement. A data-driven approach to solving the fingering problem was introduced by Radisavljevic and Driessen [24], whose algorithm learns weight costs of a particular style from fully-labelled tabs. Tuohy and Potter [25] have investigated genetic algorithms for navigating the large search space of possible fingerings for a given piece.

Hori and collaborators [21] attempted automatic fingering and arrangement using input-output Hidden Markov Models, whilst others have attempted to determine playing position directly from audio [26]–[28], sometimes constrained by a player's proficiency [29], or by using video cues as an additional input [30].

III. Structural Segmentation

Rhythm guitar compositions feature a large amount of repetition, with specific parts repeated in different realizations of the same structural part (verse, chorus etc.)[8]. To produce realistic rhythm guitar parts, the first stage of our processing pipeline is therefore a structural segmentation of the chord sequence a user inputs to our system. We employ a novelty-based approach based on the work by Foote [31].

A. Self-Similarity Matrix and Novelty Curve Computation

The input to our algorithm is a text file of M lines–one for each measure in the song. Each line represents the chords in a measure, with up to one chord symbol per sixteenth note (one chord symbol per beat or even measure can be used to reduce typographical burden for the user). Let $\mathbf{C}$ be this input chord file:

$$\mathbf{C} = [\mathbf{c}_1, \ldots, \mathbf{c}_M], \quad \mathbf{c}_i \in \mathcal{A}^{16},$$

with the chord symbols for a measure $\mathbf{c}_i$ selected from a chord alphabet $\mathcal{A}$. The first stage of processing in our method is to compute an $M \times M$ self-similarity matrix $\mathbf{S} \in [0,1]^{M \times M}$, with affinity between two measures defined by normalized Hamming similarity:

$$\mathbf{S}_{i,j} = \frac{1}{16} \sum_{t=1}^{16} \mathbb{I}(c_i^t = c_j^t), \quad i,j = 1, \ldots, M,$$

where $\mathbb{I}(\cdot)$ is an indicator function which returns 1 if and only if the input statement is true. Any pair of measures in nonequal time signatures were set to have similarity 0. An $n \times n$ binary checkerboard matrix $\mathbf{B}$ (see [31]) is then passed through the diagonal of $\mathbf{S}$, calculating the novelty at measure m as:

$$\text{Novelty}(m) = \sum_{k=m-n/2}^{m+n/2} \sum_{l=m-n/2}^{m+n/2} C_{k,l} \times B_{k,l}, \quad m = 1, \ldots, M.$$

The resulting novelty curve was subsequently normalized to [0,1] by dividing by maximum value.

B. Segment Boundary Detection

Next, we detect segment boundaries from the novelty curve. Each measure boundary can either be a boundary or not, meaning there are 2^{M-1} possible segment boundary solutions. We use a Genetic Algorithm (GA) to explore this large solution space, with a fitness function which assigns high scores to solutions which attain high novelty scores but are also sparse. Genetic algorithms have been explored in the past for structural segmentation [32], although our approach differs in that our approach is applicable to any scenario where a novelty curve is specified, including the symbolic chord and audio domains (this will be demonstrated in Section VI).

The initial population was set to 1,000 individuals with boundaries in random positions in expectation every 8 downbeats (by setting the probability of a boundary at each point to be $\frac{8}{M}$). Let the sum of the novelty function be N, and a particular individual in the population denoted by $\mathbf{x} \in \{0,1\}^M$. The fitness of an individual given the novelty curve is then defined by:

$$f(\mathbf{x}, \text{Novelty}) = \left[\frac{1}{N} \sum_{m=1}^{M} \mathbb{I}(x_m = 1) \times \text{Novelty}(m)\right] + \left[1 - \frac{1}{M} \sum_{m=1}^{M} x_m\right].$$

The first of these terms specifies how much of the total novelty is picked up by $\mathbf{x}$ and will be maximal when every downbeat is selected as a boundary. The second term in contrast rewards sparse solutions and is maximal for solutions with no boundaries. As an additional constraint to avoid solutions with unrealistically short segments, any population members with a segment less than 4 downbeats in length were assigned a fitness of 0.

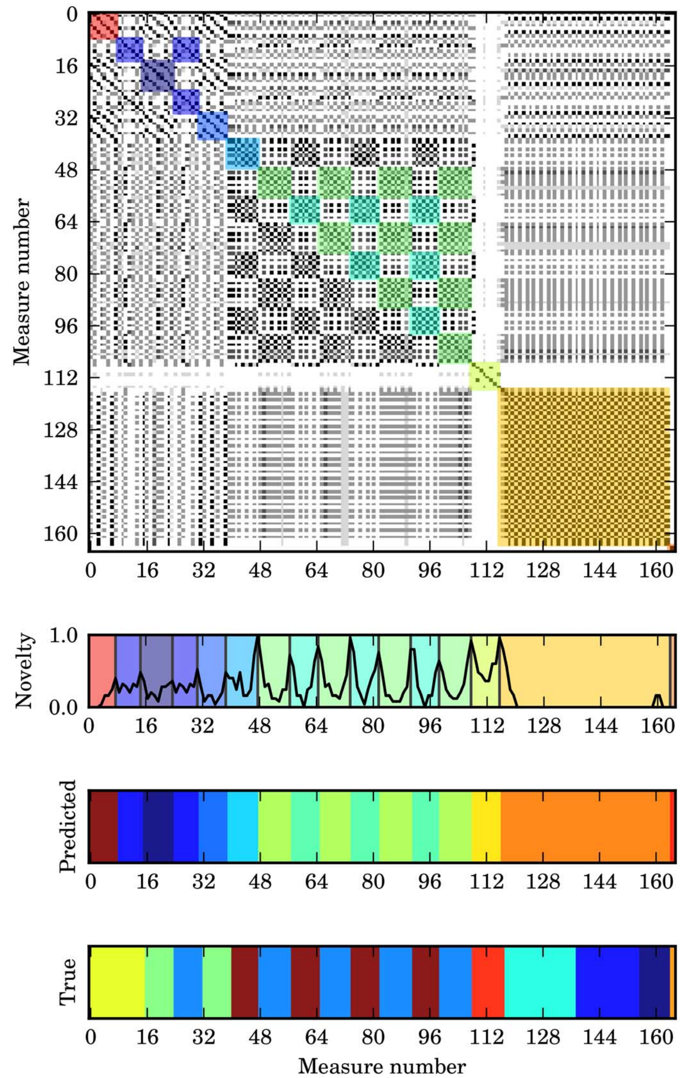


Fig. 4. Example of our symbolic structural segmentation algorithm for "Stairway to Heaven" (Page/Plant). Top: downbeat-synchronous chord sequence Self-Similarity Matrix (SSM), with darker shades indicating higher similarity. Second subfigure: novelty curve derived by passing an 8 × 8 checkerboard matrix through the diagonal of the SSM. Final two subfigures: the resulting predicted segmentation and ground truth segmentation. Throughout, sections with the same segment label have identical color.

The population was then evolved over 100 generations with single point crossover at a random position, and mutation consisting of bit swaps with probability of $\frac{1}{M}$ for each measure. Individuals for subsequent generations were chosen from the population using fitness proportionate selection, with the final boundaries chosen as the segmentation with highest fitness at generation 100.

C. Segment Label Assignment

Every segment was initially assigned a unique label. Segment pairs which were either the same length or integer multiple lengths of each other were then considered for merging (assigning the same label).

Let the number of segments identified by the GA described above be d. The similarity between each pair of segments was stored in an upper triangular matrix $\mathbf{L} \in [0,1]^{d \times d}$, with similarity of two segment pairs defined by Hamming similarity (concatenating together copies of the shorter segment to make

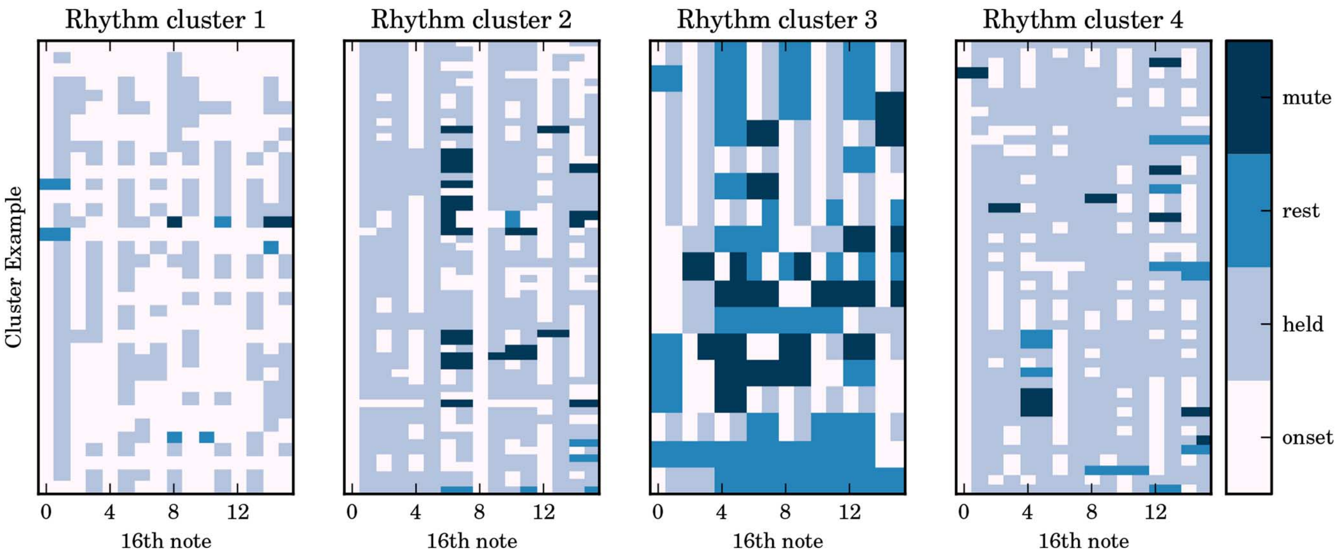


Fig. 5. Rhythm clusters derived using the techniques discussed in Section IV, with $k = 4$ clusters and trained on existing digital tablatures by 'Slash' (Saul Hudson). Each of the discovered clusters is shown in a separate subplot, with members of each cluster (example rhythms) collated in rows. Rhythm cluster 1 shows rhythms with many onsets, rhythm cluster 2 mainly has onsets on the main metrical positions (16th notes 0, 4, 8, 12), cluster 3 has many rests and muted notes, whilst cluster 4 consists mainly of held notes.

each segment of equal length if necessary). Segment pairs which were not integer multiple lengths of each other were assigned a similarity of 0.

We then inspected $\mathbf{L}_{\max}$, the largest element of $\mathbf{L}$, finding it at row i and column j. We assign segments i and j the same label if $\mathbf{L}_{\max}$ exceeds a threshold $\tau \in [0, 1]$. $\mathbf{L}_{i,j}$ is then set to 0 and we iterate until $\mathbf{L}_{\max} < \tau$.

An example of our algorithm for the chords to "Stairway to Heaven" (Page/Plant) is shown in Fig. 4, where in this example and throughout the remainder of this paper we set the parameters $n = 8$ (recall: size of cherboard matrix), $\tau = 0.5$. Here, our algorithm has inserted an erroneous boundary at measure 8 and missed two boundaries near the end of the song (due, we discovered, to constant chord sequence but changing instrumentation), but is otherwise correct. Most segment labels are correctly identified. A quantitative analysis of this algorithm is presented in Section VI-B.

IV. AutoRhythmGuitar

Our rhythm guitar generation module consists of two independent musical aspects: rhythm and pitch. The composition of these attributes is guided by the structural analysis conducted in Section III and is outlined in the following subsection. The reader is referred to our original publication on rhythm guitar generation [8] for a more thorough overview of this module.

A. *AutoRhythmGuitar Rhythm Assignment*

In addition to note onsets, rests (periods of silence) and sustained notes, guitarists are able to produce percussive *muted notes* by relaxing their fretting hand across the strings and strumming. Believing these muted notes to be important in expressive rhythm guitar playing, our rhythm model for rhythm guitar consists of the following rhythmic note states: {note onset, held note, rest, muted note}.

As in our previous work, our algorithm proceeds by clustering the training rhythms into k clusters, where k is the number of unique segment labels identified by our segmentation algorithm. The motivation for this clustering is to produce one rhythmic style for each segment type in the target song–see Fig. 5 for an example of the kinds of rhythm clusters we discovered. Each segment label is then assigned a cluster of rhythms, which are pulled from a 'bag-of-rhythms' model from the appropriate cluster in generation. Rhythms are repeated across segments with the same label for intra-segment consistency.

B. *AutoRhythmGuitar Pitch Assignment*

Our next task is to assign pitches to the note onsets determined in IV-A. We achieve this by constructing a bigram model in the tablature space. Each state in our rhythm guitar pitch model consists of a list of (string, fret, bend) triples, with bends measured in integer number of semitones. We found rhythm guitarists rarely bent notes (only 55 of 8,430 states had at least one bent string), but included bends for consistency with AutoLeadGuitar's pitch model (see V-C). An illustrative example of our state space for rhythm guitar pitch can be seen by inspecting the first eighth note in Fig. 3. In our model, this would be described as:

$$\text{state} = [(2, 2, 0), (3, 2, 0), (5, 0, 0)],$$

Adopting the convention that strings are numbered from 1 (highest-pitch) to 6 (lowest-pitch). We took a biased random walk over the state space for a chord to assign pitches from the model, exploiting the fact that any consecutive pair of states which occur in our training data are likely to be playable.

States were also transposed up and down the guitar neck to increase generalization potential of our model. The maximum transposed fret was chosen such that on no string did any state exceed the 11th fret, whilst the minimum fret was set such that the lowest note was an open string[2]. States which contained combinations of open and fretted strings were not transposed

[2]an 'open' string is one which has no fretted finger positions.

as it was not clear if transposing them would result in playable hand positions.

As noted in our previous study, sampling initial states for a chord often resulted in unrealistic and unplayable jumps around the fretboard. To counter this, we interpolated between an initial distribution for each chord and a state similarity metric, inspired by the work of Hori *et al.* [21]. We opted for a simpler state similarity in the current work, defining the similarity between two states by number of frets between the positions of the first finger.

Specifically, let the minimum (non-open string) frets in two states $\mathbf{s}_1, \mathbf{s}_2$ be $f^1_{\min}$ and $f^2_{\min}$. The similarity between $\mathbf{s}_1$ and $\mathbf{s}_2$ was then calculated via:

$$S_{\text{state}}(\mathbf{s}_1, \mathbf{s}_2) = \frac{1}{12}|f^1_{\min} - f^2_{\min}|, \tag{1}$$

where the normalisation is performed by maximum possible distance (12 frets). We then set the probability distribution of the first state for a new chord c by interpolating between the initial distribution P_{ini} for c and the similarity between the previous state $\mathbf{s}_{\text{prev}}$ and all states in the state space for c:

$$P(\cdot|c) = \alpha S_{\text{state}}(\cdot, \mathbf{s}_{\text{prev}}) + (1-\alpha)P_{\text{ini}}(\cdot|c)$$

with interpolation weight $\alpha \in [0,1]$. By setting α close to unity, we anticipate that our model will output a sequence of states which are close to each other on the fingerboard, whilst values of α closer to zero will more faithfully represent the initial state distributions for each chord. We present an analysis of the parameter α, including a method for estimating its value given labelled data, in Section VI-C.

V. AutoLeadGuitar

This section outlines our method for generating lead guitar parts. We begin by discussing the importance of phrasing in guitar solos and how this is modelled in AutoLeadGuitar. A more thorough description of AutoLeadGuitar can be found in our previous work [9].

A. AutoLeadGuitar Phrase Generation

We believe effective *phrasing* to be one of the key concepts in generating realistic lead guitar compositions. It has been shown that the composition of music in distinct phrases facilitates perception and analysis by the human auditory system [33], and is essential in conveying expressiveness in music [34].

Many improvisational instrument groups (such as those in the wind or brass family) are easily able to construct musical phrases by periodically pausing for breath. This is not the case for stringed instruments, making tasteful phrasing of improvised solos particularly challenging for guitarists. With this in mind the first stage of our lead guitar generation is the generation of phrase boundaries, which will later be populated with rhythms (Section V-B) and pitches (Section V-C).

Given a set of measures over which to play, we first choose a position within the first measure to begin playing, as well as a phrase duration in musical time. This forms the start and end of the first phrase. A phrase onset is then selected for the next phrase, chosen from whatever remains of the current measure and the next full measure. This process is then repeated until all solo measures are exhausted.

Phrase starts and durations were modelled probabilistically from data, necessitating the annotation of solo phrase starts and ends for the training data by the authors, which were rounded to the nearest sixteenth note for the purposes of the current analysis. Phrase start probabilities for each sixteenth note were estimated from histogram counts (with bin width equal to one sixteenth note), whilst phrase durations were modelled using a negative binomial distribution–the discrete analogue of the gamma distribution, commonly used to model wait times [35].

There exists no closed form for determining the maximum likelihood solutions to determining the number of trials r and stop probability p of the negative binomial distribution analytically, prompting us to determine these parameters numerically. Analytical and musicological analyses of the parameters r and p are conducted in Section VI.

B. AutoLeadGuitar Rhythm Assignment

Lead guitar rhythms were assigned by constructing a note onset bigram model, which specified the probability of an onset at each point within a phrase.

To generate rhythms for each phrase, we simply set the first onset to be the start of the phrase and took a random walk over the rhythm model until the phrase was exhausted, extending the final note to end at the start of the following phrase (as described in Section V-A–see Chapter 3 of [1] for a description of Markov-based rhythm generation). The final note of the final phrase was sustained to the end of final measure of the solo. We also added vibrato to the final note of each phrase to help easily identify phrase boundaries in our output.

C. AutoLeadGuitar Pitch Assignment

Pitch states for AutoLeadGuitar consisted of a list of (string, fret, bend) triples as per Section IV-B–note that we have increased the sophistication of this aspect of our system from our previous work to now include polyphony. We noticed that bends were far more common in lead guitar playing, with 2,884 of 6,179 states containing at least one bent string. All training data was transposed to a common key, and also transposed up and down the guitar neck up to a maximum fret of 24. When generating pitches, states were assigned by a random walk over the state space for a key.

Our previous work revealed that guiding the random walk process towards chord tones resulted in a significant increase in listener satisfaction [9]. Guiding the random walk process was implemented in this first study by interpolating between the state probabilities for a chord, and a uniform distribution which was zero everywhere except for states which were a chord tone.

To extend this idea to a polyphonic state space, we set the target distribution function to be the number of strings whose pitch was a chord tone, normalized by number of active strings in the state. For example, a state which contains a C and D note when the underlying chord is a C major was assigned a weight of 0.5. The weight was subsequently normalized over all candidate states to form a probability distribution.

As before, the interpolation weights were chosen to be $\beta \in [0,1]$ if a state occurred as the the final onset in a phrase and $\gamma \in [0,1]$ otherwise, where we suspect $\beta > \gamma$ in most cases. Methods for learning β and γ from labelled data are presented in Section VI.

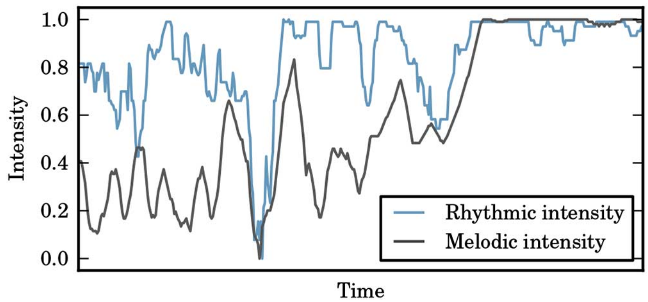


Fig. 6. Rhythmic and melodic intensity of the solo to "All Along the Watchtower" (Dylan, produced by Hendrix). Rhythmic intensity is shown in blue, melodic intensity in gray.

D. Controlling Intensity

In Section III, we conducted a structural analysis before generating rhythm guitar content, facilitating the repetition of tab content across multiple instances of the same segment label. Our intuition is that the long-term evolution of lead guitar parts relies much less on repetition of identical content across phrases and is instead guided by an evolution of 'intensity' throughout a song.

This intensity, we propose, can be controlled by the note duration and pitch height of the composition. For an example in an existing work see Fig. 6, where we have plotted the rhythmic and melodic intensity through the solo for "All Along the Watchtower" (Dylan, produced and performed by Hendrix). We defined rhythmic intensity simply as the inverse of note duration, and melodic intensity as MIDI note number. Both of these features were smoothed with a moving average of 9 notes and range-normalized.

Fig. 6 shows that the intensity of a guitar solo can vary throughout a composition. In this lead guitar part, we see that there can exist periods of negative correlation (near the start of the composition) and positive correlation (nearer the end) within a solo, suggesting to us that rhythmic and melodic intensity should be modelled separately.

In a step towards modelling intensity, users of AutoGuitarTab are able to optionally specify rhythmic and melodic *intensity contours* which guide the evolution of the tablatures generated by AutoLeadGuitar–the details of how these functions modify our system's output is outlined in the remainder of the current subsection.

Rhythmic Intensity: Let the bigram lead rhythm transition matrix formed Section V-B be denoted $\mathbf{R} \in [0,1]^{16\times 16}$. Guiding the rhythmic intensity of the lead part given an intensity contour can easily be realized by making the matrix $\mathbf{R}$ a function of intensity.

Given the previous note onset index $k \in \{1,\ldots,16\}$ and current intensity $I \in [0,1]$, we interpolate between the kth row of $\mathbf{R}$ and a function which biases towards either sixteenth notes (at points of high intensity) or eighth notes (at points of low intensity). In particular, when $I \geq 0.5$ (high intensity), we interpolate towards a delta function peaked one sixteenth note away from the previous onset, with interpolation weight equal to I. When $I < 0.5$, we interpolate towards a function which has a single peak one eighth note away from the previous onset, now with interpolation weight $(1 - I)$. Note that the model is unaltered when $I = 0.5$ (neither high nor low intensity).

Melodic Intensity: Melodic intensity was controlled by choice of first note in each phrase. The reason for this is that pitch choice within phrases is already the result of an interpolation scheme between the state distribution and chord tones (see Section V-C). Furthermore, we are optimistic that specifying the first note in a phrase is sufficient for being able to control the overall pitch intensity of a phrase (this will be investigated in VI-D).

We therefore ranked the MIDI numbers of states for each key (using the highest pitch in cases of polyphonic states) resulting in a list of sorted states $[\mathbf{s}_1,\ldots,\mathbf{s}_N]$. Given the current intensity $I \in [0,1]$ we then set the first state of the solo to be $\mathbf{s}_{\lfloor N.I \rfloor}$.

Both rhythmic and melodic intensities were set to be constant within phrases to ensure consistency within motifs. Particular intensity functions $f : \{1,\ldots,N_{\text{phrases}}\} \to I \in [0,1]$ we explore in this paper include a linear increase in intensity and a positive parabola, featuring high intensity at the start and end of a lead guitar part with a calmer section towards the middle (other more complex intensity models are naturally possible). The efficacy of our system to model intensity will be evaluated in Section VI-C.

VI. Evaluation

Evaluating the performance of algorithmic composition systems is known to be challenging, owing to the subjective nature of music quality [36]. There are however a collection of techniques which can be used to qualitatively or quantitatively assess such systems. Three particular examples of evaluation strategies we explore in this paper are algorithmic evaluation, non-expert user studies, and detailed expert studies. Each of these techniques have benefits and drawbacks, and are in particular ranked by increasing qualitative efficacy but decreasing quantitative efficacy and scalability.

In this paper we take a balanced approach, using algorithmic means to assess our model's parameters, conducting a listening test with non-expert[3] individuals, and asking a professional guitarist to assess playability and modelling of style.

In the first of these, we train models from our data and inspect the parameter values and distributions obtained. This methodology has the advantage of being easily scalable and may give us musicological insight into aspects of style learned by our system. However, this analysis gives us no idea of to what extent our system successfully produces 'realistic' music. To research this problem we conduct a medium-scale listening experiment with musical but non-expert listeners in Section VI-E, asking a number of participants to identify which of a pair of audio clips was human-generated and which was composed algorithmically.

Note however that these non-guitarists are unlikely to be able to read guitar tablature, or be familiar enough with popular music guitarists to identify nuances in stylistic playing. Finally then, to understand to what extent our system produces playable tablature and captures the style of guitarists, we asked a professional guitarist to play and record some of the output to our system, and also to try and identify the target guitarist in a number of stylistic experiments.

[3]'non-expert' in this paper means an individual with no guitar training

TABLE I
GUITARISTS, THEIR ASSOCIATED ARTISTS AND THE SONGS USED TO TRAIN AUTORHYTHMGUITAR AND AUTOLEADGUITAR

Guitarist	Associated act(s)	Songs
Eric Clapton	Solo, Cream, John Mayal & the Bluesbreakers, Derek & the Dominos	Badge, Bad Love, Cocaine, Crossroads, Hide Away, Layla, Nobody Knows You When You're Down and Out, Old Love, Sunshine of Your Love, Tears in Heaven, White Room, Wonderful Tonight
Jimi Hendrix	The Jimi Hendrix Experience	All along the Watchtower, Angel, Bold as Love, Castles Made of Sand, Come On, Crosstown Traffic, Fire, Foxy Lady, Hey Joe, Spanish Castle Magic, Stone Free, The Wind Cries Mary
Jimmy Page	Led Zeppelin	Babe I'm Gonna Leave You, Communication Breakdown, Good Times Bad Times, Houses of the Holy, Immigrant Song, Ramble On, Rock and Roll, Stairway to Heaven, Tangerine, The Lemon Song, The Rover, Whole Lotta Love
Keith	The Rolling Stones	(I Can't Get No) Satisfaction, Almost Hear You Sigh, Angie, It's Only Rock 'n' Roll, Jumping Jack Flash, Lady Jane, The Last Time, Under My Thumb, You Can't Always Get What You Want, Wild Horses
Slash	Guns N' Roses	Civil War, It's So Easy, Knockin' on Heaven's Door, Live and Let Die, My Brownstone, Nightrain, November Rain, Paradise City, Sweet Child o' Mine, Welcome to the Jungle, You Could Be Mine

The remainder of the current Section is organized as follows. In Section VI-A we outline the data we used to train our models. Section VI-B is then concerned with evaluation of our structural segmentation algorithm (for which there are well-established performance strategies). A parameter space analysis is then conduced in VI-C, before the investigation into intensity (VI-D), non-expert survey (VI-E) and expert analysis of playability and style (VI-F).

A. Data Collection

Digital guitar tabs in GuitarPro format were collected from the user-generated content site gprotab.net[4]. Five popular guitarists (Eric Clapton, Jimi Hendrix, Jimmy Page, Keith Richards, Slash) were chosen for analysis, with ten songs chosen as training data. We chose these guitarists as examples of iconic guitarists which we believe cover a range of styles which could be interesting from a pedagogical perspective. Note that, being data-driven, our system can easily be trained with new data if desired by the user.

When more than one tab was available for a song, the most accurate or complete tab was chosen (the first author was sufficiently familiar with the songs in the study that he was able to identify the most accurate version of each song and also verify the quality of this user-generated content).

We discovered that Keith Richards very rarely played lead guitar parts, and that some songs did not feature any lead guitar parts at all. Additional songs for each of the four remaining guitarists were therefore obtained in order to ensure a balanced dataset of ten rhythm and lead parts for each artist–the complete training set is shown in Table I. These tabs were subsequently converted to MusicXML to facilitate automated analysis, and annotated with downbeat-synchronized chord, key, structural segmentation labels for use in training and evaluation.

Note that our data source contains tens of thousands of GuitarPro file tablatures, although we have chosen to work with a select few in this study. The reason for this is that annotating some parts of the training data (in particular detailed chords and solo phrase boundaries) even on this small set required many hours' work. We therefore decided to work with a set of around 50 tracks where we could ensure high-quality data and annotations. Despite this, we found that by creating pitch-invariant models we actually had many data/training points. For example, we found that our model contained:

- 4,425 rhythm guitar rhythm measures (809 of which were unique)
- 940 unique major chord states, 6,254 state-to-state transitions (both after transposition)
- 210 unique lead guitar phrases, consisting of 6,128 individual notes
- 2,726 unique lead guitar melodic states, 17,957 state-to-state transitions (after transposition)

Naturally, using more data from our source would lead to more powerful compositional models. However, we feel that we have gathered and annotated enough data in the current study to make models of reasonable quality. One option we plan to explore in the future is automatic or semi-automatic means of annotating more data which may allow us to exploit a larger number of digital tabs, at the cost of data quality.

B. Structural Segmentation

We evaluated our structural segmentation algorithm on two datasets. First, we evaluated in the symbolic domain on the data from Table I, using ground truth chord sequences annotated at the beat level. This experiment will assess how well our segmentation algorithm performs in the most realistic use case, when a user inputs a chord sequence as per Fig. 1. True chords and beat structure are used under the assumption that users correctly input them into our system. Next, we evaluated on a standard dataset in the audio domain, estimating chords and downbeat positions using automated methods. Although this is not directly applicable to usage in our system, it will give an indication of how well our algorithm performs with imperfect chord sequences and downbeats estimated from audio.

Structural segmentation performance in both cases was evaluated with the Boundary Detection and Frame Pair Clustering metrics [37]. Precision, recall and $f-$measure were computed for each of these metrics. Since in the symbolic domain there are no explicit onset points, we required an exact match on structural boundaries for a 'hit'. The results of these experiments, broken down by artist, can be seen in the top portion of Table II.

[4]http://www.guitar-pro.com/en/index.php, http://www.gprotab.net/

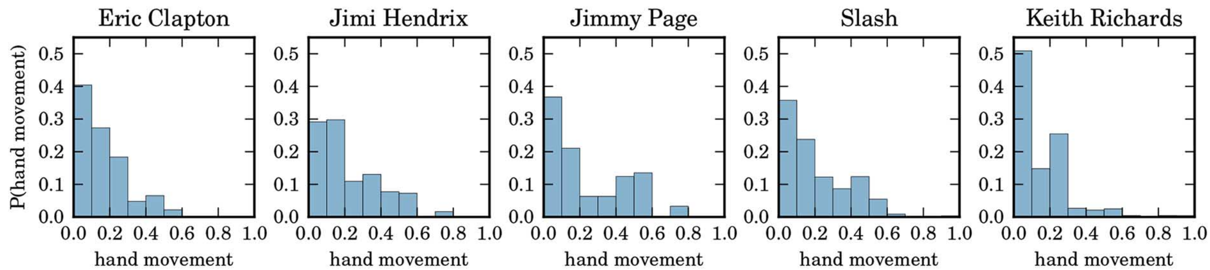


Fig. 7. Fretting hand movements for five guitarists (normalized by maximum fretting hand movement of 12 frets), trained from existing performances.

TABLE II
STRUCTURAL SEGMENTATION ALGORITHM PERFORMANCE. ABBREVIATIONS: p: PRECISION, r:RECALL, f: $f-$ measure

	Boundary Detection			Frame Pair Clustering		
Guitarist	p	r	f	p	r	f
Eric Clapton	0.51	0.52	0.50	0.54	0.43	0.45
Jimi Hendrix	0.63	0.72	0.67	0.83	0.55	0.67
Jimmy Page	0.64	0.72	0.67	0.69	0.53	0.57
Keith Richards	0.53	0.59	0.55	0.74	0.43	0.53
Slash	0.50	0.60	0.54	0.66	0.53	0.57
Mean of above	0.56	0.63	0.59	0.69	0.49	0.55
RWCpop [38]	0.28	0.33	0.28	0.54	0.36	0.42
MIREX [41], [42]	0.55	0.38	0.43	0.53	0.62	0.56

Inspecting Table II, we see that our algorithm achieves a good balance between precision and recall of boundaries (mean precision 0.56, mean recall 0.63). In Frame Pair Clustering however, we achieved higher precision than recall (mean precision 0.69 compared to mean recall 0.49), suggesting that our algorithm was too cautious in assigning two segments the same label. Interestingly, performance varied across artists, suggesting certain harmonic structures are easier or more challenging for our system to detect than others. Inspecting individual songs, we discovered our algorithm performed most impressively when the chord sequence accurately reflected the song structure (as expected). Specific examples of when this assumption failed included blues songs (whose repetitive chord sequence failed to reveal any novelty) and the instrumental sections of songs (which our algorithm was unable to distinguish from sections with the same underlying chord sequence).

Next, we evaluated on a standard audio dataset to assess if our symbolic structural segmentation algorithm could be extended to work with audio input and to compare to existing methods. To facilitate this, we extracted downbeats and chords from audio for the RWC pop music dataset (RWC-MDB-P-2001 Nos. 1-100, hereafter referred to as 'RWCpop') [38], using the web service Songle [39]. Chords were then downbeat-synchronized and converted to the format required by AutoRhythmGuitar and fed into our structural segmentation algorithm. Evaluation was performed using the `mir_eval` library [40] with a $0.5s$ window for Boundary Detection and can be seen in the seventh line of Table II (labelled 'RWCpop'). We also include the two best-performing algorithms (ranked by boundary and Frame Pair Clustering) from the 2014 MIREX evaluation for direct comparison (final row of Table II, labelled 'MIREX').

Comparing these results to those above, we see lower magnitude Boundary Detection $f-$ measure for both our system ('RWCpop') and the cutting-edge ('MIREX'). This is to be expected as under these experimental conditions the feature extraction is conducted on audio, which contains information not explicitly relevant to the harmony (drum signals etc). In Frame Pair Clustering we achieved an $f-$ measure of 0.42 compared to the cutting-edge of. This can be explained we believe by our choice of fitness function in the genetic algorithm we outlined in III-B, which takes into account novelty and position of boundaries but not segment labels.

C. Parameter Space Analysis

Rhythm Guitar Fretting Hand Movement: The parameter α introduced in IV-B represents how strongly a rhythm guitarist wishes to minimize fretting hand movement when changing chords. We approximate α by gathering fretting hand movement from existing data when chords change and sampling from the resulting distribution. Specifically, each time a new chord was presented, we calculated the distance a guitarist moved as a proportion of the maximum movement (12 frets). These distances were then binned into a histogram with leftmost bin edges at $\{0.0, 0.1, \ldots, 0.9\}$. These counts were subsequently normalized and are shown (per guitarist) in Fig. 7.

From this Figure, it is clear that fretting hand movement between chords is an artist-specific trait. Specifically, although all guitarists' movements are peaked towards minimal mobility, the degree to which they move their fretting hand differs. For example, Keith Richards appears to be the most economical player, moving a normalized distance of less than 0.1 (corresponding to 1 fret) in over half of all chord changes, whilst Jimi Hendrix exhibits a fatter-tailed distribution.

It is apparent from Fig. 7 that a single value of α is not appropriate for modelling fretting hand movement. Therefore, each time a chord change is encountered we sample an alpha value from the centres of the bins defined above. We tested the significance of the differences between values of β for pairs of guitarists and found at least one pair which was significantly different ($p < 0.05$) for each guitarist. For γ, the same test yielded tail probabilities $p < 10^{-10}$, indicating higher significance. With more guitarists in future work we are interested to see if the difference between β and γ are significant, although the limited sample size of 4 lead guitarists limits the power of this test in the current scenario.

Lead Guitar Phrase Parameters: Fig. 8 shows the phrase onset and duration parameters derived from our data for each lead guitarist. The former is represented by a histogram over each sixteenth note, the latter by maximum likelihood estimation of a negative binomial distribution. Again, it is clear from these plots that guitarists exhibit individuality with respect

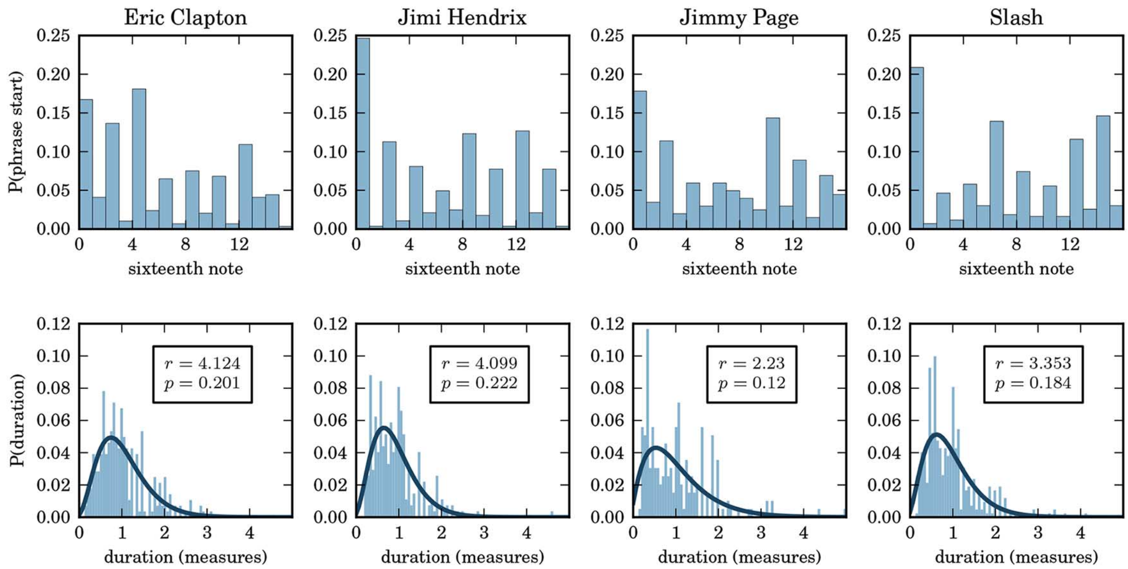


Fig. 8. Maximum likelihood estimates for lead guitar phrase parameters for each training artist. Top row: phrase start duration, in sixteenth note resolution. Lower row: phrase durations modelled by negative binomial distributions. Bars in the plots in the second row show individual data points (phrase durations), dark curves indicate the maximum likelihood estimate parameters of the negative binomial distribution for each artist (parameters also shown in legend).

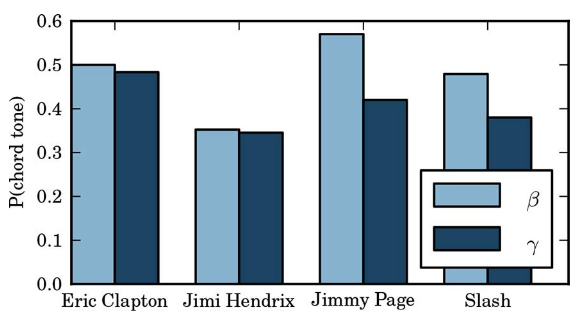


Fig. 9. End of phrase β and mid-phrase γ chord tone preferences for the lead guitarists estimated from our dataset.

to this aspect of their playing. All guitarists generally favour phrases to start on eighth notes, but for example Jimi Hendrix begins solo phrases on the first sixteenth note in approximately 1 in every 4 phrases. Eric Clapton on the other hand most often starts phrases on beat 2.

Regarding phrase duration, the peak for all guitarists is a little under 1 measure, but the exact shape of the distribution is guitarist-specific. Phrase durations for Jimmy Page are particularly interesting as they seem to have high variance–many phrases are less than 1 measure long but also a significant number are longer than 3 measures.

Lead Guitar Chord Tone Preference: In the AutoLeadGuitar system, the parameters β and γ represent preferences for lead guitar parts selecting a chord tone during phrases (γ) and at the end of phrases (β). We estimated these parameters by simply counting the number of times a lead guitarist selected a state which had at least one chord tone (discriminating between mid-phrase and end-phrase notes). The normalized counts for each guitarist can be seen in Fig. 9.

Most striking to us from Fig. 9 was the similarity in β and γ for Eric Clapton and Jimi Hendrix. These values suggest that neither of these players use chord tones more frequently on the final notes of their phrases, counter to our intuition. Jimi Hendrix also generally plays chord tones less frequently than other guitarists. Jimmy Page and Slash do however tend to end their lead phrases on chord tones ($\beta > \gamma$). The other guitarists in our dataset ended phrases on chord tones more often than in the middle of phrases as we expected.

D. Intensity

Recall that we allow users of our system to optionally input intensity curves describing the desired evolution of the rhythmic and melodic intensity of the generated solos. To assess the effect these curves had on the output to our system, we therefore generated 10 guitar solos for a fixed song ('Stairway to Heaven', Page/Plant) with no intensity curve specified, and 10 with a linear increase in rhythmic intensity and parabolic function of melodic intensity. We then calculated the intensities of the resulting solos and plotted them in Fig. 10.

From this Figure, we see no clear patterns in the top two subfigures. However, the bottom subfigures show that by setting intensity contours (linear increase in rhythmic intensity, parabolic function of melodic intensity) we were indeed able to steer the global properties of the solo towards user preference. We find it especially interesting that the melodic intensity can be controlled so effectively simply by specifying the start note of each phrase–this tells us that guitarists tend to play each phrase in a particular pitch range. Videos demonstrating our ability to control intensity are available on our YouTube page.[5].

E. Non-Expert Listener Tests

A Turing test was conducted with amateur musicians working in the field of music information retrieval to assess the ability of our system to generate realistic guitar parts. Six participants were presented with two pairs of audio clips per artist and AutoGuitarTab module, one of which came from the training data and the other from our system.

Audio was synthesized using GuitarPro with a multi-instrument backing track (to give musical context) with all guitar parts other than the part of interest removed. Clip length, start and

[5]https://youtube.com/channel/UCdQYTz0-qiHq4UCuQVnJp4g

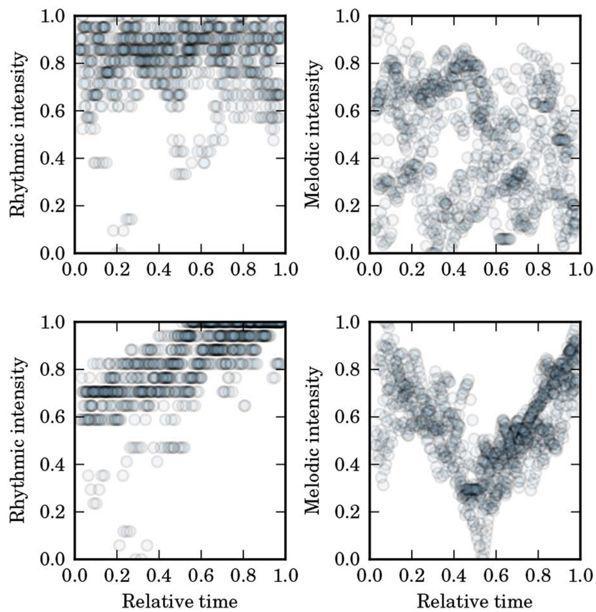


Fig. 10. Melodic and Rhythmic intensities of 20 guitar solos generated by AutoLeadGuitar. In the top two subfigures no intensity curve was specified, whilst in the bottom a linear increase in rhythmic increase in intensity and parabolic function of melodic intensity was desired.

TABLE III
TURING TEST PERFORMANCE ACROSS AUTO GUITAR TAB MODULES AND TRAINING ARTISTS. DATA INDICATE THE NUMBER OF TIMES THE HUMAN-GENERATED COMPOSITION WAS CORRECTLY IDENTIFIED

	Module/correctly identified		
Guitarist	Rhythm	Lead	Total
Eric Clapton	8/12	3/12	11/24
Jimi Hendrix	9/12	10/12	19/24
Jimmy Page	9/12	11/12	20/24
Slash	3/12	10/12	13/24
Keith Richards	11/12	-	11/12
Total	40/60	34/48	74/108

end time and guitar tone were standardized across clip pairs and presentation order was also randomized. Results for this experiment can be seen in Table III.

The second column of Table III shows that our listeners were able to correctly identify the human rhythm guitarist in just 40 of 60 cases. We find this result encouraging, given the difficulty of the task. We tested the hypothesis that the distribution of correct answers came from a binomial distribution with probability of success equal to 0.5 (i.e. that the listeners were guessing randomly) and found the resulting p-value to be 0.043. This suggests that our algorithm has composed music which is to some extent indistinguishable from human-generated compositions.

For lead guitar parts (third column of Table III) 34/48 guitar solos were correctly identified as human-generated, resulting in a p-value of 0.015. We attribute the relative difficulty of the Turing tests compared to rhythm guitar playing to the improvised nature of guitar solos and their use of medium-term repetition such as repeated 3 or 4 note motifs.

F. Expert Analysis of Style

In our final two analyses, we asked an expert (professional guitarist with more than 10 years teaching experience) to assess the output of our system. First, we generated rhythm and lead guitar parts in the style of each guitarist and asked him to try and identify the source artist, using tablature alone (i.e., no cues from guitar tone, backing track etc.). Results for these experiments can be seen in Table IV.

TABLE IV
CONFUSION MATRICES FOR EXPERT IDENTIFICATION OF STYLE

		Guitarist				
Module	**Guitarist**	C	H	P	S	R
AutoRhythmGuitar	Eric (C)lapton	**3**	0	0	0	0
	Jimi (H)endrix	0	**5**	0	0	0
	Jimmy (P)age	0	0	**1**	2	2
	(S)lash	2	0	1	**1**	1
	Keith (R)ichards	0	0	2	1	**2**
AutoLeadGuitar	Eric (C)lapton	**0**	3	1	0	–
	Jimi (H)endrix	1	**1**	0	2	–
	Jimmy (P)age	1	0	**1**	2	–
	(S)lash	2	0	2	**0**	–

From Table IV, we see that 12 of 25 tablatures generated by AutoRhythmGuitar were correctly classified by our expert. The probability of witnessing an event at least this extreme under the null hypothesis (the expert was assigning guitarists randomly) was calculated to be less than 10^{-10}, indicating with high confidence that we should reject the null hypothesis and that AutoRhythmGuitar is able to model style. We found it particularly interesting that rhythm guitar parts generated in the style of Jimi Hendrix were all correctly identified by our expert. A demonstration video showing the different styles were are able to model is available online[6].

As per Section VI-E, we noticed that results for Lead guitar were less impressive. Specifically, only 2 of 16 AutoLeadGuitar parts were correctly identified by our expert. The conclusion we may draw from this is that either lead guitar playing is not style-dependent, or our model has failed to capture whichever elements of lead guitar playing constitute style.

We were interested to see which features our expert used to classify rhythm guitar examples, especially for the Hendrix tracks, which were identified without error. When prompted, he responded that he found certain artists easy to identify:

> "*The rhythm tracks were pretty interesting, I found that some jumped out instantly as certain guitarists*"

Specifically, he commented that Hendrix *"... rarely takes a simple chord and strum approach"* and that *"often he's a bit more rhythmically complex than most the other players"* and used these to identify this artist in our tests. In lead playing, he commented that actually, symbolic music data might not be sufficient for identifying players:

> "*I think with solos it often isn't just melodic content that is the noticeable difference but literally how Clapton plays his vibrato or how Page plays his bend and repeat phrases as how I'd identify this*"

This difference between the *spectrographic* and *calligraphic* aspects of style has been noted in the literature [43], and is beyond the scope our symbolic music generation model.

[6]https://www.youtube.com/watch?v=JYL3QLqqclU

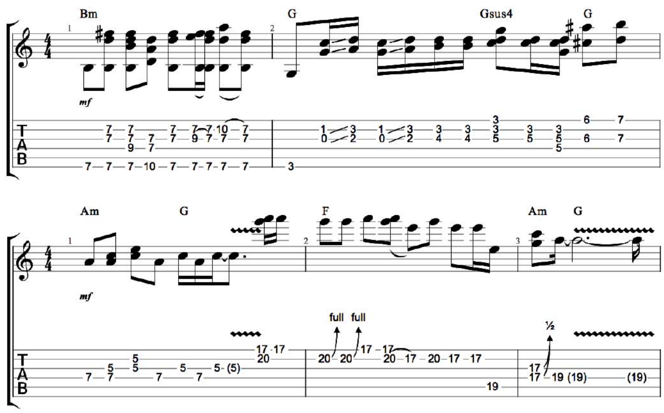


Fig. 11. Output from AutoGuitarTab. Top stave: AutoRhythmGuitar output for 'Wild Horses' (Jagger/Richards) in the style of Jimi Hendrix. Bottom stave: AutoLeadGuitar output for 'Stairway to Heaven' (Page/Plant) in the style of Eric Clapton. Phrase ends in the bottom figure are indicated with vibrato.

G. Playability and Individual Composition Analysis

Finally, recall that one of the main motivations for our system generating tablature was to maintain playability. This aspect of our system was tested by asking our expert to play two tablatures generated by our system–one for rhythm guitar (in the style of Hendrix), the other for lead (in the style of Slash). We also asked our expert for his detailed opinion on the style and playability of these songs. Videos showing the generated content with his recording overlaid are available online[7].

For rhythm guitar, our expert commented that the piece was lacking some of the properties he was expecting from a Hendrix composition. In particular he said it was lacking arpgeggiated chords, pentatonic phrasing during rhythm sections and walking basslines. He did however mention that some of the chord voicings were very accurate:

> "*Some of the ways that the chords were voiced did seem on target, with the thumb over the neck approach on some of the chords.*"

For overall structure, he felt that there was too much repetition (*"It was also a fairly straight forward and repetitive rhythm, which you don't really expect from Hendrix"*). This interestingly indicates that our assumption of repetition of rhythm guitar content throughout the piece is not appropriate for this player. He also found the piece to be playable: *"From a playing point of view there wasn't a lot of issues once I'd got to used to the thumb over the neck approach"*, although naturally he found some of the bars more challenging than others (bar number 64 in particular). Style was modelled better in the lead guitar part according to our expert:

> "*This seemed to have more of the technical and melodic stylings that I'd expect from a Slash piece. The unison bends were exactly what I'd expect and the repetitive bends with a syncopated rhythm.*"

However, our experts' detailed knowledge of Slash's playing style revealed that some of the positions would be challenging given Slash's instrument of choice: (*"The placement for some of the notes were odd (e.g. the intro on a D string) This would be awkward on a Les Paul and would sound better on possibly a B or G string"*). He also noted the lack of a coherent intensity through the piece (we did not specify an intensity contour in this composition):

> "*It also fizzles out at the end, the crescendo really feels like its in bars 138-9*"

This we feel is excellent evidence that modelling of intensity is necessary for generating realistic guitar solos. Finally we note that overview videos are available on our YouTube page, and show brief example output from our system in Fig. 11.

VII. Conclusions

In this paper, we introduced a statistical method for guitar composition and improvisation. Our model composes playable, structurally-consistent rhythm guitar parts and solos in distinct phrases which highlight chord tones and were guided in intensity by user input. Altering the training data of our model allowed us to generate artist-specific models.

Our main contributions in this work were advancements in the evaluation of AutoRhythmGuitar and AutoLeadGuitar, but also a new structural segmentation algorithm which is applicable to both symbolic and audio data, and the introduction of intensity curves which effectively steer the global rhythmic and melodic properties of lead guitar parts.

In future work, we would like to improve our GA for Boundary Detection to include segment label information (including more musically-inspired fitness functions), incorporate repeated melodic motifs into our generation system, and investigate the relative importance of score, microtimings and guitar tone (choice of instrument, amplifier etc.) on artist identification performance in listening experiments.

Acknowledgment

We would like to thank the anonymous reviewers for their insightful comments.

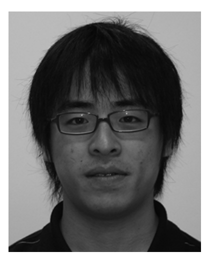
References

[1] S. Nierhaus, *Algorithmic Composition*. New York, NY, USA: Springer Wein, 2009.
[2] L. Hiller and L. Isaacson, "Musical composition with a high speed digital computer," in *Audio Engineering Society Convention*. New York, NY, USA: Audio Engineering Soc., 1957.
[3] F. Pachet, "The continuator: Musical interaction with style," *J. New Music Res.*, vol. 32, no. 3, pp. 333–341, 2003.
[4] A. Addessi and F. Pachet, "Experiments with a musical machine: Musical style replication in 3 to 5 year old children," *British J. Music Educat.*, vol. 22, no. 1, pp. 21–46, 2005.
[5] J. Shepared, *Continuum Encyclopedia of Popular Music of the World*. Edinburgh, U.K.: A&C Black, 2003, ch. 9.
[6] C. Ames, "The Markov process as a compositional model: A survey and tutorial," in *Leonardo*, 1989, pp. 175–187.
[7] J. Abeßer, "Automatic string detection for bass guitar and electric guitar," in *Proc. 10th Int. Symp. Computer Music Modelling and Retrieval*, 2013, pp. 333–352.
[8] M. McVicar, S. Fukayama, and M. Goto, "AutoRhtyhmGuitar: Computer-aided composition for rhythm guitar in the tab space," in *Proc. Joint Int. Comput. Music Conf. and Sound and Music Comput. Conf.*, 2014, pp. 293–300.
[9] M. McVicar, S. Fukayama, and M. Goto, "AutoLeadGuitar: Automatic generation of guitar solo phrases in the tablature space," in *Proc. Int. Conf. Signal Process.*, 2014, pp. 599–604.
[10] H. Olson, *Music, physics and engineering*. New York, NY, USA: Courier Dover Publications, 1967, vol. 1769.
[11] L. Hiller and R. Baker, "Computer cantata: A study in compositional method," *Perspect. New Music*, pp. 62–90, 1964.

[7]https://www.youtube.com/watch?v=n9_e8n9h_RU

[12] I. Xenakis, *Formalized music: thought and mathematics in composition*. Hillsdale, NY, USA: Pendragon, 1992, no.6.
[13] R. Zaripov and J. Russell, "Cybernetics and music," in *Perspect. New Music*, 1969, pp. 115–154.
[14] D. Conklin and I. Witten, "Multiple viewpoint systems for music prediction," *J. New Music Res.*, vol. 24, no. 1, pp. 51–73, 1995.
[15] D. Ponsford, G. Wiggins, and C. Mellish, "Statistical learning of harmonic movement," *J. New Music Res.*, vol. 28, no. 2, pp. 150–177, 1999.
[16] D. Conklin, "Music generation from statistical models," in *Proc. Symp. Artif. Intell. Creativity in the Arts Sci.*, 2003, pp. 30–35.
[17] F. Pachet, P. Roy, and G. Barbieri, "Finite-length markov processes with constraints," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2001, pp. 635–642.
[18] F. Pachet and A. Addessi, "When children reflect on their own playing style: Experiments with continuator and children," *Comput. Entertain. (CIE)*, vol. 2, no. 1, pp. 14–32, 2004.
[19] N. Chomsky, *Syntactic Structures*. Berlin, Germany: Walter de Gruyter, 2002.
[20] A. Papadopoulos, P. Roy, and F. Pachet, "Avoiding plagiarism in Markov sequence generation," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2731–2737.
[21] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *J. Inf. Process.*, vol. 21, no. 2, pp. 264–271, 2013.
[22] S. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Comput. Music J.*, vol. 13, no. 3, pp. 76–84, 1989.
[23] D. Radicioni, L. Anselma, and V. Lombardo, "A segmentation-based prototype to compute string instruments fingering," in *Proc. Conf. Interdisc. Musicol.*, 2004.
[24] A. Radisavljevic and P. Driessen, "Path difference learning for guitar fingering problem," in *Proc. Int. Comput. Music Conf.*, 2004, vol. 28.
[25] D. Tuohy and W. Potter, "GA-based music arranging for guitar," in *Proc. IEEE Congr. Evol. Comput.*, 2006, pp. 1065–1070.
[26] A. Barbancho, A. Klapuri, L. Tardon, and I. Barbancho, "Automatic transcription of guitar chords and fingering from audio," *Trans. IEEE Audio, Speech, Lang. Process.*, vol. 20, no. 3, pp. 915–921, Mar. 2012.
[27] Y. Yazawa, D. Sakaue, K. Nagira, K. Itoyama, and H. Okuno, "Audio-based guitar tablature transcription using multipitch analysis and playability constraints," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 196–200.
[28] E. Humphrey and J. Bello, "From music audio to chord tablature: Teaching deep convolutional networks to play guitar," in *Proc. Int. Conf. Acoustics, Speech and Signal Processing*, 2014, pp. 7024–7028.
[29] Y. Yazawa, K. Itoyama, and H. Okuno, "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 3122–3126.
[30] A. Burns and M. Wanderley, "Visual methods for the retrieval of guitarist fingering," in *Proc. Int. Conf. New Interfaces for Musical Express.*, 2006, pp. 196–199.
[31] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *Proc. Multimedia and Expo*, 2000, vol. 1, pp. 452–455.
[32] B. Rafael, S. Oertl, M. Affenzeller, and S. Wagner, "Optimization of parameter settings for genetic algorithms in music segmentation," *Comput. Aided Syst. Theory*, pp. 240–247, 2012.
[33] Y. Nan, T. R. Knösche, S. Zysset, and A. Friederici, "Cross-cultural music phrase processing: An fMRI study," *Human Brain Mapping*, vol. 29, no. 3, pp. 312–328, 2008.
[34] K. Jensen and O. Khl, *Towards a Model of Musical Chunks*, ser. Lecture Notes in Computer Science. Berlin/Heidelberg, Germany: Springer, 2009, vol. 5493.
[35] R. Hogg and A. Craig, *Introduction to Mathematical Statistics*. London, U.K.: Macmillan, 1978, ch. 3.
[36] G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: A survey, a critical view and future prospects," in *Proc. AISB Symp. Musical Creativity*, 1999, pp. 110–117.
[37] H. Lukashevich, "Towards quantitative measures of evaluating song segmentation," in *Proc. 9th Int. Soc. Music Inf. Retrieval*, 2008, pp. 375–380.
[38] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Popular, classical and jazz music databases," in *Proc. 3rd Int. Conf. Music Inf. Retrieval*, 2002, pp. 287–288.
[39] 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. 12th Int. Soc. Music Inf. Retrieval*, 2011, pp. 311–316.
[40] C. Raffel, B. McFee, E. Humphrey, O. Nieto, D. Liang, and D. Ellis, "mir-eval: A transparent implemenation of common MIR metrics," in *Proc. 15th Int. Soc. Music Inf. Retrieval*, 2014, pp. 367–372.
[41] J. Schluter, K. Ullrich, and T. Grill, "MIREX submission: Structural segmentation with convolutional neural networks," Austrian Research Inst. for Artif. Intell., Tech. Rep, 2014.
[42] O. Nieto and J. Bello, "MIREX submission: 2D Fourier magnitude coefficients," Music and Audio Research Lab, New York Univ., Tech. Rep., 2014.
[43] F. Benadon, "Spectrographic and calligraphic cues in the identification of jazz saxophonists," in *Proc. 5th Triennial ESCOM Conf.*, 2003, pp. 246–249.

Matt McVicar received a Master of Science in mathematics in 2008 and Master of Research in complexity sciences in 2009, both from the University of Bristol. Just prior to the completion of his Ph.D. at the same institution, he was awarded a scholarship by the USUK Fulbright commission to conduct research at Columbia University in the city of New York. After this he worked at the National Institute for Advanced Industrial Science and Technology in Tsukuba, Japan before returning to the University of Bristol as a Research Associate in data science. His research interests are focused on data-driven applications to Music Information Retrieval tasks, and in particular the use of freely available internet resources to solve these problems.



Satoru Fukayama received his Ph.D. degree in information science and technology from the University of Tokyo in 2013. He is currently a Research Scientist at the National Institute of Advanced Industrial Science and Technology (AIST) in Tsukuba, Japan. His main interests are in automated content generation (music compositions, music performances and choreographies) with probabilistic modeling, machine learning and computational musicology. He has received several awards including the IPSJ Yamashita SIG Research Award from the Information Processing Society of Japan.

Masataka Goto received the Doctor of Engineering degree from Waseda University in 1998. He is currently a Prime Senior Researcher and the Leader of the Media Interaction Group at the National Institute of Advanced Industrial Science and Technology (AIST), Japan. Over the past 23 years, he has published more than 220 papers in refereed journals and international conferences and has received 40 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, the Tenth JSPS PRIZE, and the Commendation for Science and Technology by the Minister of Education, Culture, Sports, Science and Technology (Young Scientists' Prize). He has served as a committee member of over 90 scientific societies and conferences, including the General Chair of the 10th and 15th International Society for Music Information Retrieval Conf.s (ISMIR 2009 and 2014). In 2011, as the Research Director he began a 5-year research project (OngaCREST Project) on music technologies, a project funded by the Japan Science and Technology Agency (CREST, JST).