AutoLeadGuitar: Automatic Generation of Guitar Solo Phrases in the Tablature Space

Matt McVicar, Satoru Fukayama, Masataka Goto

Abstract—**We present AutoLeadGuitar, a system for automatically generating guitar solo tablatures from an input chord and key sequence. Our system generates solos in distinct musical phrases, and is trained using existing digital tablatures sourced from the web. When generating solos AutoLeadGuitar assigns phrase boundaries, rhythms and fretboard positions within a probabilistic framework, guided towards chord tones by two user-specified parameters (chord tone preference during and at the end of phrases). Furthermore, guitar-specific ornaments such as hammer-ons, pull-offs, slides and string bends are built directly into our model. Listening tests with our model output confirm that the inclusion of chord tone preferences, phrasing, and guitar ornaments corresponds to an increase in user satisfaction.**

Keywords—Computer generated music, Music information retrieval

I. Introduction

THIS paper introduces a method of algorithmic composition for the guitar. Guitar parts in popular music can broadly be split into *rhythm* (outlining the main harmony/rhythmic pulse) and *lead* parts (harmonies/solo breaks). Naturally extending our previous work [1], the current paper is focused on the latter.

A. Motivation

Our main motivation for this work is that automatically generated guitar solos have the potential be used in situations when a popular music composer would like a guitar solo in a piece but lacks the necessary familiarity with the instrument to compose one. Furthermore, we believe that automatically generated content could be used as a pedagogical aid, to help amateur musicians learn different approaches to playing over a given chord sequence. This work could therefore be considered an example of 'Creative MIR' [2], whose goal is to use real-world applications to transfer Music Information Retrieval results beyond the immediate research community.

B. Challenges and proposed solutions

There are several challenges which must be overcome in the automatic generation of guitar solos. First, algorithmic melody composition (of which we consider the current work to be a subtask) has been said to be challenging to evaluate without the presence of a musical context [3]. Inspired by previous work [4], [5], our system counters this by using a key and chord sequence as a user-defined input, and guides the melody towards chord tones to enhance musicality and realism.

All authors are at the National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan and can be contacted at: mattjamesmcvicar@gmail.com, s.fukayama@aist.go.jp, m.goto@aist.go.jp.

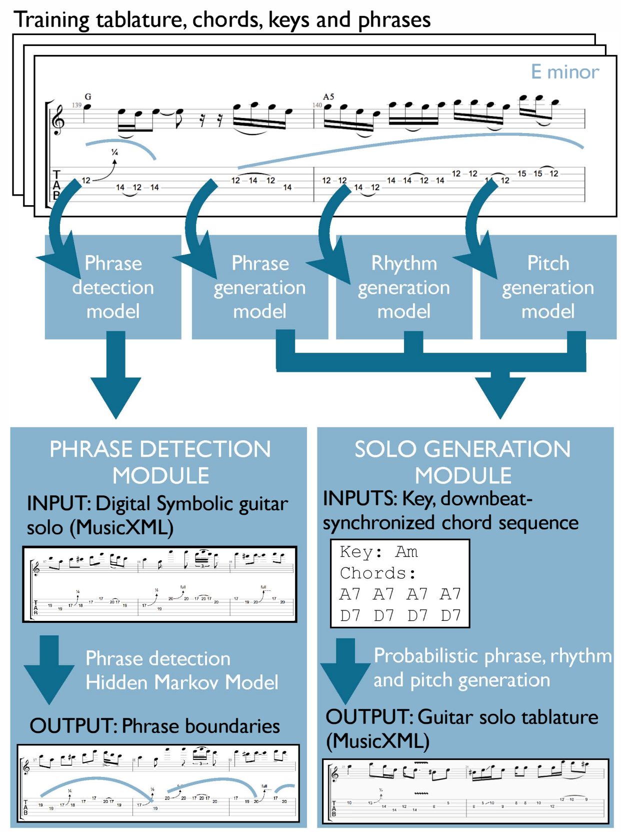

Fig. 1. Flowchart of AutoLeadGuitar's main processes. Training data (digital tablatures annotated with chords, keys and phrases) are used to create models for the detection of phrases in existing solos, and for the generation of novel solos when complemented with an input chord and key sequence.

We believe one of the key concepts in generating realistic guitar solos to be effective *phrasing*. Indeed, it has recently been shown that composing musical events in phrases facilitates perception and analysis by the human auditory system [6], and is essential in conveying expressiveness in music [7]. Phrasing is particularly challenging for guitarists when compared to other solo instruments (such as the wind instrument family) as guitarists cannot rely on pauses for breath to construct lyrical phrases. To better understand the phrasing techniques used by lead guitarists, the first stage of the current work is therefore an analysis of existing guitar solos, leading to a statistical model for the detection of phrases in digital scores. An algorithm for the generation of guitar solo phrases is subsequently introduced, defined by parameters learned from an existing corpus.

978-1-4799-2186-7/14/$31.00 ©2014 IEEE

Generating music for the guitar algorithmically is also particularly challenging owing to the physical layout of the instrument. Specifically, the pitch ranges for guitar strings significantly overlap, meaning there is more than one position (string and fret number) to play most notes. Tablature notation (or simply *tab*, plural *tabs*), which explicitly notates the position in which to play each note in a piece, was developed to overcome this ambiguity. Existing compositional models which output traditional notation rather than tab would therefore necessitate an automatic fingering or arrangement (see II-B). To combat this and to exploit attractive aspects of the instrument such as string bends, hammer-ons, pull-offs, and slides, our model composes directly in the tablature space. A high-level outline of our proposed techniques is shown in Figure 1.

C. Paper structure

The remainder of this paper is organised as follows. In Section II we survey the literature on computer-aided composition, algorithmic guitar fingering/arrangement and phrase boundary detection. Section III outlines our phrase boundary detection and compositional model, which are evaluated in Section IV. Finally, we conclude the work and discuss areas of future research in Section V.

II. Relevant literature

A. Computer-aided melody generation

Algorithmic composition has a rich and varied research history (see, for example, [8], [9] and the survey [10]), of which computer-aided composition [5], in which the compositional task is split between human and computer, is an interesting subtask. The interaction between human expert and the machine in computer-aided composition may occur in a fixed order, or involve several iterations back-and-forth [11].

The generation of melodies given a chord sequence is a particularly popular topic in computer-aided composition, a particular example of which is John Biles' GenJam [4], a genetic algorithm for the generation of jazz solos. GenJam takes a chord sequence as input and outputs an improvised solo, with melody phenotype fitness optimized over several iterations via user feedback. Pachet and various collaborators [12], [13] have tackled generation of sequences (including melodies) via Markov processes with hard constraints such as "end the phrase on this note" and with great care taken to avoid plagarism in model output.

B. Automatic guitar fingering and arrangement

As mentioned in Section I, the layout of the guitar means that mapping a given musical score to string and fret positions is non-trivial, and may not even be possible. Algorithmically mapping a score to a tab and minimally altering a score to ensure it is playable on the guitar are referred to as algorithmic *fingering* and *arrangement* respectively.

Sayegh first considered the problem of algorithmic fingering for stringed instruments [14], introducing an *optimum path paradigm* algorithm, later developed by Radicioni [15] to minimise fingering difficulty at the phrase, rather than global level. A data–driven approach to the same problem was suggested by Radisavljevic and Driessen [16], whose *path difference learning* learns the weight costs of a particular playing style based on labelled tabs. Genetic algorithms have also been explored as a means of efficiently exploring the large search space created in the fingering decision problem [17]. Hori et al. [18] designed an input-output Hidden Markov Model (HMM) for the automatic arrangement of guitar pieces.

C. Automatic phrase boundary detection

Existing phrase boundary detection methods are mostly based in the audio domain. Aucouturier and Sandler [19] detected changes in spectral similarity to identify phrase boundaries, whilst Cheng and Chew [20] instead used loudness and expressive parameters. An unsupervised HMM was presented by Kim and Weinzierl [21] which used cues including note duration relative to the preceding note. Pearce et al. conducted a comparison and evaluation of symbolic data-driven and rule-based phrase detection algorithms [22], finding that a hybrid model was able to attain an $f-$measure of 0.66.

III. Methods

A. Identification of guitar solo phrase boundaries

We formulate phrase boundary detection probabilistically via a supervised Hidden Markov Model, with observed states as symbolic note properties (pitch, note duration etc.) and states corresponding to the underlying phrase structure.

When inspecting existing solos, we noticed that some phrases contained brief rests and that in others phrases ran consecutively without rest, meaning that the presence of a rest is neither necessary nor sufficient to segment phrases. Furthermore, we noticed that phrase ends had distinct characteristics which made them amenable to automatic identification, most notably that they tended to end on chord tones and on strong metrical position, and had duration significantly longer than their predecessors. On the basis of these observations, our model has three hidden states at each time point t: $x_t \in \{\text{no phrase}, \text{phrase}, \text{phrase end}\}$ (note that phrase starts can be uniquely defined as following immediately from a 'no phrase' or 'phrase end').

Hidden chain model parameters for the HMM (initial distribution of phrases $P_{\text{ini}}(x_1)$ and phrase-to-phrase transitions $P_{\text{trans}}(x_t|x_{t-1})$) were set using a fully-labelled dataset using maximum likelihood estimation (MLE). We set the observation probabilities $P_{\text{obs}}(y_t|x_t)$ based on the following assumptions. First, that phrases are unlikely to contain rests. Second, phrases end on strong metrical positions and harmonically stable pitches. Finally, phrase ends are likely to have a duration significantly longer than their predecessors.

To model these properties mathematically, we extracted the following properties of each symbolic note: whether the note was an onset or a rest, the metrical position (quantized to sixteenth notes), the pitch relative to the underlying chord, and note duration relative to its predecessors (calculated by taking the ratio of the note duration and the median duration of the previous three notes). We then set the probability of witnessing note y given state x at time t as the product of y's rest, metric position, relative duration, and chord tone probabilities:

$$P_{\text{obs}}(y_t|x_t) = P_r(y_t|x_t)P_m(y_t|x_t)P_d(y_t|x_t)P_c(y_t|x_t).$$

P_r, P_m and P_c were estimated from training data directly from normalised histogram counts. P_d represents a waiting time and is therefore naturally modelled as a Gamma distribution, the shape and scale parameters of which were estimated using MLE. The joint probability of a hidden state sequence $\mathbf{x} = (x_1, \ldots, x_T)$ and observed note sequence $\mathbf{y} = (y_1, \ldots, y_T)$ given these parameters can then be computed via:

$$P(\mathbf{x}, \mathbf{y}) = P_{\text{ini}}(x_1) \prod_{t=2}^{T} P_{\text{trans}}(x_t|x_{t-1}) \cdot P_{\text{obs}}(y_t|x_t)$$

The state sequence which maximises this quantity:

$$\mathbf{y}^* = \underset{\mathbf{y}}{\text{argmax}}\, P(\mathbf{x}, \mathbf{y})$$

can then found efficiently via the Viterbi algorithm [23].

B. Generation of phrase boundaries

Our phrase generation algorithm is intuitive and we believe mimics the creative process guitarists use when improvising a solo. It is outlined in Figure 2. Given a set of measures over which to play, our algorithm begins by choosing a metric position in the first measure (phrase onset selection, A) and a phrase duration (phrase duration selection, B). A phrase onset is then chosen for the next phrase (A), chosen from whatever remains of the current measure and the next full measure. This process is then repeated until the solo measures are exhausted. We implemented this algorithm by calculating a phrase start probability P_{ps} and phrase duration distribution P_{pd} from our training data. We found that P_{pd} was well approximated by a normal distribution, with mean usually a little over one measure. Phrase onset probabilities carried over multiple measures (such as the second iteration of algorithm stage A in Figure 2) were normalised to meet the probability criterion.

C. Generation of rhythms

Phrase rhythms were set from a note onset bigram model, which specified the probability of an onset at each sixteenth note given the last onset position. These probabilities were calculated from our training data and collected into a matrix:

$$\mathbf{R} \in \mathbb{R}^{16 \times 16}, \text{ such that } \sum_{j=1}^{16} \mathbf{R}_{i,j} = 1.0 \quad \text{for } i = 1, \ldots, 16.$$

The above-diagonal elements of $\mathbf{R}$ represent transitions further into the current measure, elements below and on the diagonal represent transitions into the following measure. The first onset of each phrase was set to be start of the phrase itself, with a random walk over the rows of $\mathbf{R}$ used to generate subsequent onsets within the phrase, until the phrase was exhausted.

In practice we found that favourable results were obtained by extending the offset of the final note of a phrase to the onset of the following phrase, increasing the duration of the final note in the phrase and mimicking the behaviour seen in existing solos (recall Subsection III-A).

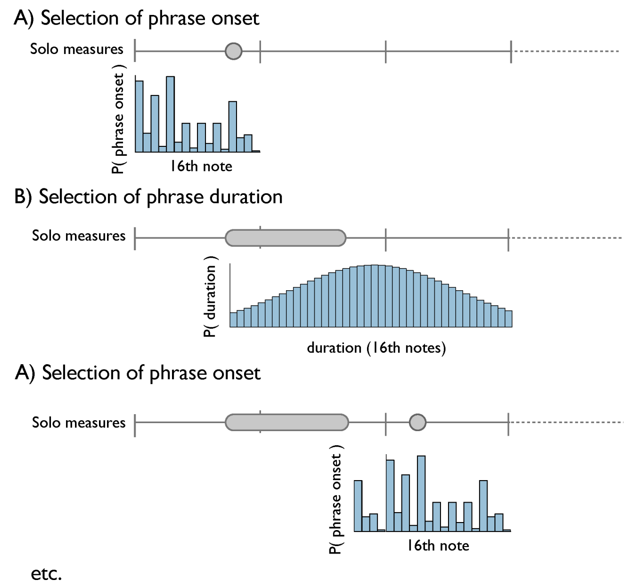

Fig. 2. AutoLeadGuitar's phrase generation algorithm, which alternates between phrase onset selection (stage A) and phrase duration selection (stage B) until the designated measures are exhausted.

D. Phrase pitch state space:

To set the pitch for each note onset, we constructed a bigram model in the tablature space. To incorporate string bending, a facet we believe to be important in expressive guitar playing, each unique state in the pitch model consisted of a (string, fret, bend) triple. Bends were measured in integer number of semitones (0 for unbent states). This data was collected in a matrix $\mathbf{P} \in \mathbb{R}^{|S| \times |S|}$, where $|S|$ denotes the number of unique (string, fret, bend) state triples seen in the data.

E. Key normalization

To maximally exploit our training data and increase our system's generalisation potential, we transposed our data into two canonical keys (C major/A minor) before normalising the rows of $\mathbf{P}$. The reason we believe a key dependent, rather than chord dependent model to be valid is that lead guitarists in the blues and rock genre frequently improvise using the ubiquitous pentatonic scale (pitch classes A, C, D, E, G in the key of C major/A minor), as observed by Flor and Holder [24]:

"For rock and roll and blues songs, knowledge of pentatonic scales and a song's key is usually sufficient for improvising."

$\mathbf{P}$ was 'untransposed' into a user-specified key during testing.

F. Highlighting chord tones

We found that by simply taking random walks over $\mathbf{P}$, at the end of phrases our model frequently ended on dissonant or unresolved tones, resulting in a slightly frustrating listening experience. Similarly, we noticed that interesting non-diatonic chord tones were unfortunately rarely highlighted. One method to counter this suggested to students of the guitar is to highlight specific chord tones in solos:

"...try starting your idea on a chord and aim for another, meeting the change with a strong, chord-defining pitch." [25]

We incorporated this behaviour in our compositions by added two parameters to guide the random walk process towards chord tones. Specifically, given the current pitch state s_t we calculate the probability of the next state s_{t+1} by interpolating between the t-th row of $\mathbf{P}$, and a function which is 1.0 if s_{t+1} is a chord tone and 0.0 otherwise:

$$P(s_{t+1}|s_t) = w\mathbf{P}(s_{t+1}|s_t) + (1-w)\mathbb{1}(s_{t+1} \in c_{t+1}),$$

where c_{t+1} are the pitch classes of the chord at time $t+1$ and $w \in [0, 1]$ is an interpolation weight. Recalling from Subsection III-A that many phrases ended on chord tones, we set $w = \beta$ when the note was the final note in a phrase and γ otherwise, where $\beta, \gamma \in [0, 1]$ and generally $\beta > \gamma$.

G. Exploiting guitar-specific ornaments

When two subsequent states occur on the same string, a guitarist has the option of transitioning between them in at least three novel ways. Briefly defining some terminology: a *slide* between states is a simple glissando from one fret to the next, a *hammer-on* is sounded by 'hammering' from one fret on a string to a higher fret on the same string without plucking/picking, and a *pull-off* is formed by 'pulling' one finger from the fretboard onto a lower fret.

The probability of any one of these ornaments occurring between any two states in the model was learnt from data using the same methodology above. We stored the probabilities in (fairly sparse) key normalized hammer-on/pull-off and slide transition matrices $\mathbf{HP}, \mathbf{S} \in \mathbb{R}^{|\mathcal{S}| \times |\mathcal{S}|}$ for each ornament. Note that hammer-ons and pull-offs may be stored in a single matrix since they are only possible on state transitions strictly increasing/decreasing fret numbers respectively. A hammer-on/pull-off between states with indices i and j was then included with probability $\mathbf{HP}_{i,j}$. If no hammer-on or pull-off was included, a slide was between the states was added with probability $\mathbf{S}_{i,j}$.

IV. Experiments

A. Data collection

We collected digital guitar tablatures to train our model from the user-generated content website GuitarPro.net[1], choosing ten songs each from four popular guitarists in the blues/rock style: Jimi Hendrix, Eric Clapton, Jimmy Page, and 'Slash' (Saul Hudson). Guitarists were chosen according to tab popularity (number of tabs), with songs chosen which were mostly in common time and in standard tuning (or tuned sharp or flat 1 semitone, which is easily transposed). When more than one tab was available for a song, the most accurate or complete tab was chosen. Each tab was then converted to MusicXML format via GuitarPro for automated analysis.

[1] http://www.gprotab.net/

TABLE II. Phrase boundary detection performance. Columns 2-4: inter-artist testing. Columns 5-7: intra-artist testing.

	Performance					
	Inter-artist			Intra-artist		
Guitarist	p	r	f	p	r	f
Eric Clapton	0.74	0.61	0.67	0.81	0.55	0.66
Jimi Hendrix	0.79	0.55	0.65	0.79	0.50	0.61
Jimmy Page	0.78	0.58	0.66	0.82	0.64	0.72
Slash	0.73	0.47	0.57	0.70	0.52	0.59
Total	0.73	0.54	0.63	0.70	0.54	0.63

Downbeat-synchronised chord sequences, key sequences and phrase boundaries were then annotated for each solo by the authors. In total we found our dataset to contain 1,266 guitar solo phrases consisting of 11,854 individual notes.

B. HMM for guitar solo phrase boundary detection

We tested our HMM for phrase boundary detection using cross-fold validation. Unsure if the guitarists in our training set exhibited artist-specific phrasing, we performed two types of experiment. First, for each artist we held out one test song and trained our model on the remaining songs, repeating this for each of the artist's ten songs (*inter-artist* testing). We were also interested to see if a more general phrase detection model could be developed, and so for each test song also trained a model on every tab which was not by the given test artist (30 tabs total, *intra-artist* testing).

Performance was measured by calculating the precision, recall, and $f-$measure of detection of phrase boundaries, with an exact match required for a 'hit'. The results of our experiments can be seen in Table II. Inspecting the left portion of Table II, we see that the total precision of our model in detecting boundaries was 0.73. Recall was lower than precision, implying that our model was too cautious in predicting boundaries. In $f-$measure our boundary detection performance totals 0.63, a figure comparable with existing methods [26]. Interestingly, the right portion of Table II reveals similar performances, indicating that the features we used for detecting phrases are shared among guitarists, and that a general model for detecting phrase boundaries in symbolic guitar solos is plausible.

Upon closer inspection of our results, we found the reason for lower recall in boundary detection was that our model did not predict a boundary when confronted with short repeated melodic motifs. These cues are currently not built into our detection or generation model, with the knock-on consequence that our generated solos will not exhibit this behaviour. Building repetition into our boundary detection and generation models therefore forms part of our future work.

C. Generation of novel guitar solos

The quality of the generated solos was assessed by the use of listening tests. We studied the efficaciousness of the three main attributes of our compositional model: highlighting chord tones, musical phrasing, and guitar-specific ornaments,

TABLE I. GUITARISTS, THEIR ASSOCIATED ARTISTS AND THE SONGS USED TO TRAIN THE MODEL.

Guitarist	Associated acts	Training tabs
Eric Clapton	Solo, Cream, Derek & the Dominos, John Mayall & the Bluesbreakers	Bad Love, Badge, Crossroads, Cocaine, Hide Away, Layla, Nobody Knows You When You're Down and Out, Old Love, Sunshine of Your Love, White Room, Layla
Jimi Hendrix	The Jimi Hendrix Experience	All Along the Watchtower, Bold as Love, 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, Rock and Roll, Stairway to Heaven, Tangerine, The Lemon Song, The Rover, Whole Lotta Love
Slash	Guns N' Roses	Civil War, It's So Easy, Knockin' on Heaven's Door, Mr Brownstone, Nightrain, November Rain, Paradise City, Sweet Child O' Mine, Welcome to the jungle, You Could Be Mine

testing the hypothesis that the addition of these attributes will correlate positively with listener satisfaction.

Participant and model description: Participants consisted of six researchers from the authors' institution. All were male, and had musical experience spanning 0–25 years.

Models for each of the artists in Table I were trained from nine of ten songs, with the final song held out for testing. For each artist, AutoLeadGuitar then generated four solos using the methods detailed in Subsections III-B–III-G.

To minimise the burden on participants, instead of generating solos corresponding to every possible pair of attribute activations, we generated four audio clips with increasing model complexity for each artist. Specifically, we constructed a baseline model where we set the parameters $\beta = \gamma = 0$, and 'turned off' phrasing by enforcing that the entire solo was comprised of one uninterrupted phrase. Furthermore, we limited the state space to states which featured no bends, and set the matrices $\mathbf{HP}$ and $\mathbf{S}$ to be filled with 0. In the next model we set $\beta = 0.5$ and $\gamma = 0.3$, subsequently included phrasing and finally allowed string bends and other ornaments for the most complex model.

Experimental conditions: As AutoLeadGuitar exports to MusicXML, the output can be easily synthesized in existing software packages. Audio was synthesized using GuitarPro with a generic rock guitar timbre and a full backing track. A 20s clip was taken from the middle of each of the solos with a 2s fade in and out. Presentation order was randomized across model complexity to prevent any confounding effects of familiarity, and participants were simply asked to rank the solos by each artist by preference, from 1 (favourite) to 4 (least favourite). Repeated listens of each clip was permitted.

Videos of the synthesized output (from which the 20s clips were generated) for our most sophisticated model can be viewed online [2]. An example tab produced by our system with all attributes active is also shown in Figure 3.

[2]https://vimeo.com/100385330, https://vimeo.com/100385331, https://vimeo.com/100385332, https://vimeo.com/100385333

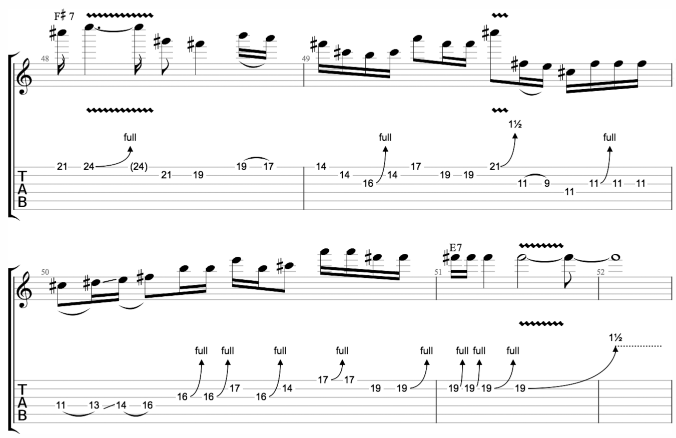

Fig. 3. Example AutoLeadGuitar output in traditional (upper) and tab (lower) notation, trained using Hendrix tabs. Hammer-ons/pull-offs are indicated by slur, slides with a slash and bends with arrows, phrase ends by vibrato.

Results: Ranks for each artist and participant obtained from our experiments can be seen in Table III. Model sophistication decreases down rows, with participant rankings shown in the rightmost columns. Kendall's τ, which counts the number of correctly ranked pairs in a list [27], was used to measure the performance of each individual ranking and shown in the final row of each artist.

All rankings in our experiments were non-negative, meaning that no participant perceived there to be a negative correlation between model sophistication and enjoyment. In half of all cases the participants ranked the clips either in complete agreement with model complexity or swapped a single pair of clips (12/24 audio clips with $\tau \geq 0.67$). The solos generated using Jimi Hendrix tabs as training data appeared to correlate most strongly with model complexity, with an average τ of 0.67 and all but one listener ranking the most sophisticated model as their favourite. Eric Clapton proved the most challenging artist to rank. An interesting effect we had not anticipated is that some participant's ranks more closely correlated with model sophistication than others (compare τ for participants B and

TABLE III. RANKINGS AND PERFORMANCE (KENDALL'S τ, GRAY ROWS) OF GENERATED SOLOS. ABBREVIATIONS: "CHRD." = CHORD TONE PREFERENCE, "PHRS." = RHTYHMIC PHRASING, "ORN." = GUITAR-SPECIFIC ORNAMENTS. CORRECT RANKINGS ARE IN BOLD.

				Participant/rank					
Artist	Chrd.	Phrs.	Orn.	A	B	C	D	E	F
Clapton	✓	✓	✓	2	**1**	3	2	**1**	2
	✓	✓	-	3	**2**	1	1	**2**	1
	✓	-	-	4	4	4	4	4	4
	-	-	-	1	3	2	3	3	3
τ				0.0	0.67	0.0	0.33	0.67	0.33
Hendrix	✓	✓	✓	**1**	2	**1**	**1**	**1**	**1**
	✓	✓	-	**2**	1	**2**	3	3	3
	✓	-	-	**3**	**3**	4	2	2	4
	-	-	-	**4**	**4**	3	**4**	**4**	2
τ				1.0	0.67	0.67	0.67	0.67	0.33
Page	✓	✓	✓	**1**	**1**	3	2	**1**	3
	✓	✓	-	**2**	**2**	**2**	3	**2**	1
	✓	-	-	**3**	**3**	1	1	**3**	2
	-	-	-	**4**	**4**	**4**	**4**	**4**	**4**
τ				1.0	1.0	0.0	0.33	1.0	0.33
Slash	✓	✓	✓	2	**1**	2	2	**1**	2
	✓	✓	✓	3	**2**	3	3	**2**	3
	✓	-	-	1	**3**	4	1	**3**	1
	-	-	-	**4**	**4**	1	**4**	**4**	**4**
τ				0.33	1.0	0.0	0.33	1.0	0.33

C). In future work and with more judges we would like to investigate if these agreements are correlated with either generic or guitar-specific musical training.

Finally, we tested the hypothesis that each successive model attribute (chord tone preference, phrasing, guitar-specific ornaments) offered a significant increase in user satisfaction by use of the one-sided Wilcoxon signed-rank test. Each increase in complexity was found to be significant at the 5% level, the only exception being the baseline model against the model with chord tone preferences ($p = 0.093$). All non-consecutive models pairs (e.g. baseline model vs. most sophisticated model) were found to offer increases in user enjoyment ($p \leq 0.001$).

V. CONCLUSIONS AND FUTURE WORK

In this paper we described AutoLeadGuitar, a system for generating phrased guitar solos given a chord and key progression. To gain insight into guitar phrasing, we designed an HMM for the detection of phrase boundaries which exceeded 0.6 in $f-$measure and was found to be artist-independent. We then generated solos in the tablature space, highlighting chord tones, using phrasing and exploiting guitar-specific ornaments. Listening experiments revealed that adding musical attributes resulted in an increase in user satisfaction, with significant differences in model rankings found in all but one case. In future work, we would like to incorporate repeated melodies into our phrase detection and generation algorithms and use a higher-resolution rhythm model.

ACKNOWLEDGEMENTS

The authors would like to thank the participants who took part in the experiment and Graham Percival for his insightful comments. This work was supported in part by CREST, JST.

REFERENCES

[1] M. McVicar, S. Fukayam, and M. Goto, "AutoRhythmGuitar: Computer-aided composition for rhythm guitar in the tab space," in *ICMCSMC (to appear)*, 2014.

[2] E. Humphrey, D. Turnbull, and T. Collins, "Creativity in music informatics," in *ISMIR late-breaking news and demos*, 2013.

[3] G. Papadopoulos and G. Wiggins, "AI methods for algorithmic composition: a survey, a critical view and future prospects," in *AISB Symposium on Musical Creativity*, 1999, pp. 110–117.

[4] J. Biles, "Genjam: A genetic algorithm for generating jazz solos," in *ICMC*, 1994, pp. 131–131.

[5] F. Pachet, "The continuator: Musical interaction with style," *Journal of New Music Research*, vol. 32, no. 3, pp. 333–341, 2003.

[6] Y. Nan, T. 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.

[7] K. Jensen and O. Khl, *Towards a Model of Musical Chunks*, ser. LNiCS. Springer Berlin Heidelberg, 2009, vol. 5493.

[8] L. Hiller and L. Isaacson, *Experimental Music; Composition with an electronic computer*. Greenwood Publishing Group, 1979.

[9] D. Cope, "A computer model of music composition," in *Machine models of music*. MIT Press, 1992, pp. 403–425.

[10] C. Ames, "The Markov process as a compositional model: a survey and tutorial," *Leonardo*, pp. 175–187, 1989.

[11] G. Nierhaus, *Algorithmic Composition*. Springer-Verlag/Wien, 2009.

[12] F. Pachet, P. Roy, and G. Barbieri, "Finite-length Markov processes with constraints," in *Artificial Intelligence*. AAAI Press, 2011, pp. 635–642.

[13] A. Papadopoulos, P. Roy, and F. Pachet, "Avoiding plagarism in Markov sequence generation," in *AAAI*, 2014.

[14] S. Sayegh, "Fingering for string instruments with the optimum path paradigm," *Computer Music Journal*, vol. 13, no. 3, pp. 76–84, 1989.

[15] D. Radicioni, L. Anselma, and V. Lombardo, "A segmentation-based prototype to compute string instruments fingering," in *Conference on Interdisciplinary Musicology*, 2004.

[16] A. Radisavljevic and P. Driessen, "Path difference learning for guitar fingering problem," in *ICMC*, vol. 28, 2004.

[17] D. Tuohy and W. Potter, "A genetic algorithm for the automatic generation of playable guitar tablature," in *ICMC*, 2005, pp. 499–502.

[18] G. Hori, H. Kameoka, and S. Sagayama, "Input-output HMM applied to automatic arrangement for guitars," *Journal of Information Processing*, vol. 21, no. 2, pp. 264–271, 2013.

[19] J. Aucouturier and M. Sandler, "Segmentation of musical signals using hidden markov models," *Audio Engineering Society*, 2001.

[20] E. Cheng and E. Chew, "A local maximum phrase detection method for analyzing phrasing strategies in expressive performances," in *Mathematics and Computation in Music*, 2009, pp. 347–353.

[21] T. Kim and S. Weinzierl, "Phrase boundary estimation in music performance with HMM-based unsuperised learning," in *ICMC*, 2012.

[22] M. Pearce, D. Müllensiefen, and G. Wiggins, "A comparison of statistical and rule-based models of melodic segmentation." in *ISMIR*, 2008.

[23] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[24] N. Flor and B. Holder, "Hearing with the eyes: A distributed cognition perspective on guitar song imitation," in *Annual Conference of the Cognitive Science Society*, 1996, pp. 148–153.

[25] M. Dziuba, *The Big Book of Jazz Guitar Improvisation*. Afred Publishing Co., Inc., 2003, ch. 5, p. 62.

[26] M. T. Pearce and G. Wiggins, "Evaluating cognitive models of musical composition," in *International joint workshop on computational creativity*, 2007, pp. 73–80.

[27] B. McFee and G. Lanckriet, "Metric learning to rank," in *ICML*, 2010, pp. 775–782.