SPARSE CEPSTRAL AND PHASE CODES FOR GUITAR PLAYING TECHNIQUE CLASSIFICATION

Li Su, Li-Fan Yu and Yi-Hsuan Yang
Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan
lisu@citi.sinica.edu.tw, a999frank@gmail.com, yang@citi.sinica.edu.tw

ABSTRACT

Automatic recognition of guitar playing techniques is challenging as it is concerned with subtle nuances of guitar timbres. In this paper, we investigate this research problem by a comparative study on the performance of features extracted from the magnitude spectrum, cepstrum and phase derivatives such as group-delay function (GDF) and instantaneous frequency deviation (IFD) for classifying the playing techniques of electric guitar recordings. We consider up to 7 distinct playing techniques of electric guitar and create a new individual-note dataset comprising of 7 types of guitar tones for each playing technique. The dataset contains 6,580 clips and 11,928 notes. Our evaluation shows that sparse coding is an effective means of mining useful patterns from the primitive time-frequency representations and that combining the sparse representations of logarithm cepstrum, GDF and IFD leads to the highest average F-score of 71.7%. Moreover, from analyzing the confusion matrices we find that cepstral and phase features are particularly important in discriminating highly similar techniques such as pull-off, hammer-on and bending. We also report a preliminary study that demonstrates the potential of the proposed methods in automatic transcription of real-world electric guitar solos.

1. INTRODUCTION

The use of various instrumental techniques is essential in music. A practical, interpretable automatic transcription system should provide information about playing techniques in addition to information about pitch or onset. For example, various fingering styles of the guitar, such as pull-off, hammer-on or bending, are all important elements of a guitar performance. A novice guitar player might be eager to learn the playing techniques employed in a musical excerpt of interest. Similar to some popular online automatic chord recognizer (e.g. Chordify [1]), a tool transcribing the note-by-note playing techniques of a guitar recording enhances the interactivity of music learning or listening experiences, and thereby offers important educational, recreational and even cultural values.

While extracting the pitch, onset, chord and instrument information from a musical excerpt has received great attention in the music information retrieval (MIR) community [3, 5, 16–18, 24], relatively little effort has been invested in transcribing the playing technique of instruments [23]. In addition, due to the use of various guitar tones (i.e. audio effects such as distortion, reverb, delay, and chorus effect) in everyday guitar performances, conventional timbre descriptors extracted from the spectrum might not be enough in modeling the electric guitar playing techniques. For instance, as the chorus effect is usually implemented by temporal delay [6], information about the phase spectrum might be important. On the other hand, for distortions that involve a filtering effect, cepstral features might be useful to characterize the respective source and filter components [8].

Motivated by the above observations, we present in this paper a comparative study evaluating the accuracy of playing technique classification of electric guitar using a variety of spectral, cepstral and phase features. The contribution of the paper is three-fold. First, to investigate more subtle variation of musical timbre, we compile an open dataset of 7 playing techniques of electric guitar, covering a variety of pitches and 7 tones (cf. Section 4). We have made the full dataset and its detailed information available online. [2] Second, as feature learning techniques such as dictionary learning and deep learning have garnered increasing attention in audio signal processing [12, 18, 22, 25], we evaluate the performance of sparse representations of audio signals using a dictionary adapted to the signals of interest (Section 5). Our evaluation shows that, to better model the playing techniques, it is useful to combine the sparse representation of different types of features, such as logarithm cepstrum and phase derivatives (Section 6). Finally, a preliminary study using a guitar solo demonstrates the potential of the proposed methods in automatic guitar transcription (Section 7).

2. RELATED WORK

Designing useful musical timbre descriptors has been a long-studied topic, and has achieved high performance in some fundamental problems such as instrument classifica-

[1] http://chordify.net/

[2] http://mac.citi.sinica.edu.tw/GuitarTranscription

tion of monophonic signals [13]. Nowadays, researchers turn to more challenging problems like multiple instrument recognition, which deals with a highly complicated timbre space [10]. Besides the complexity of multiple instruments, another challenge in timbre classification is to identify all the styles of timbre that one instrument can produce, such as to identify the playing techniques of an instrument. For exmaple, Abeßer *et al.* and Reboursière *et al.* [1, 21] pioneered the problem of automatic guitar playing technique classification, and used timber descriptors such as spectral flux, weighted phase divergence, spectral crust factors, brightness, and irregularity, amongst others. Most of these features are physically related to the characteristics of a plucked, vibrating string. However, these studies were not evaluated using a dataset comprising of various playing techniques and guitar tones.

In addition to larger and more realistic datasets, novel feature learning techniques might be helpful for modeling subtle timbre variations. Recently, sparse coding (SC) as a feature learning technique has been shown effective for MIR. This approach uses a predefined dictionary (codebook) to encode the prominent information of a given low-level feature representation of an input signal. One can encode any sensible audio representation by SC to capture different signal characteristics. For instance, Nam *et al.* [17] applied SC on short-time mel-spectra for music auto-tagging; Yu *et al.* [25] applied SC on logarithm cepstra and power-scale cepstra for predominant instrument recognition. Our work goes one step further and exploits phase information for SC.

3. ELECTRIC GUITAR PLAYING TECHNIQUE

Table 1 lists the 7 playing techniques we consider in this work. Most guitar solos are constructed with these techniques. For example, muting is widely used alternatively in place of normal in guitar *riffs* for rhythmic and punched phrases in rock and metal music, and bending is commonly considered to be the most important technique for expressing emotion.

To gain more insights into the signal-level properties of the playing techniques, in Fig. 1 we show the spectrograms (the first row) and the short-time cepstra (the second row) of the individual-note examples played with the 7 playing techniques. The first three columns are individual notes F4 of normal, vibrato and mute, the fourth column the consecutive notes F4–E4 of pull-off, and the last three columns the consecutive notes F4–#F4 of hammer-on, sliding and bending. The length of all samples is 0.6s. The window size is 46ms and the hop size is 10ms. From the spectrograms and the short-time cepstra, we see that muting has a 'noisier' attack and a faster decay comparing to normal. Moreover, hammer-on, sliding and bending have quite different transition behaviors, although they have the same note progression. The transition is sharp for hammer-on; smooth for bending; and there is a two-stage transition for sliding. Therefore, it seems that both the spectrogram and the cepstra contain useful information that can be exploited for automatic classification.

Technique	Description	# clips
Normal	Normal sound	2,009
Muting	Sounds muted (by right hand) to create great attenuation	385
Vibrato	Trilled sound produced by twisting left hand finger on the string	637
Pull-off	Sound similar to normal but with the smoother attack created by pulling off the string by left hand finger	525
Hammer-on	Sound similar to normal but with the smoother attack created by hammering on the string by left hand finger	581
Sliding	Discrete change to the target note with a smooth attack by left hand finger sliding through the string	1,162
Bending	Continuous change to the target note without an apparent attack by bending the string by left hand fingers	1,281

Table 1. Description of the playing techniques considered.

4. DATASET

While there is no publicly available dataset for guitar playing technique classification across different tones, we establish our own one with the aforementioned 7 playing techniques. The dataset is recorded by a professional guitarist using a recording interface, PreSonus' AudioBox USB, with bit depth of 24 bits and frequency response from 14 Hz to 70 kHz. We directly line-in the guitar to recording interface to catch every nuance of sound and exclude environmental noise. The guitar for recording is ESP's MII with Seymour Duncan's pickup and Ebony finger board, which is a high-quality guitar especially for metal and rock music. To make the quality of the sound recordings akin to that of real-world performance, we augment the single clean tone source to different guitar tones, which is done in the post-production stage using music production software Cubase. In addition, we assign each audio clips to 7 different guitar tones, which involve different levels of *distortion*, *reverb*, *delay* and *chorus*. Such tones may represent different genres such as rock, metal, funk, and country music solos. Moreover, the tones are carefully tuned to meet the quality for listening.

Because of the different characteristics of the techniques, the clips are recorded in slightly different ways. All the clips of sliding and bending have 2 notes for each clip with both whole step (2 semitones) and half step (1 semitone); all the clips of hammer-on and pull-off have 2 notes with only half step; and the clips of vibrato and muting have only one note for each clip. As for normal, we record whole steps, one steps, and single notes to cover all possible cases which might occur in the other 6 techniques. For sliding and bending, we record the clips only with the first three strings of the guitar since these techniques are less frequently applied on the last 3 strings. Similarly, we record muting clips with only the last 3 strings because it is commonly used in rhythm guitar with low pitch. Other

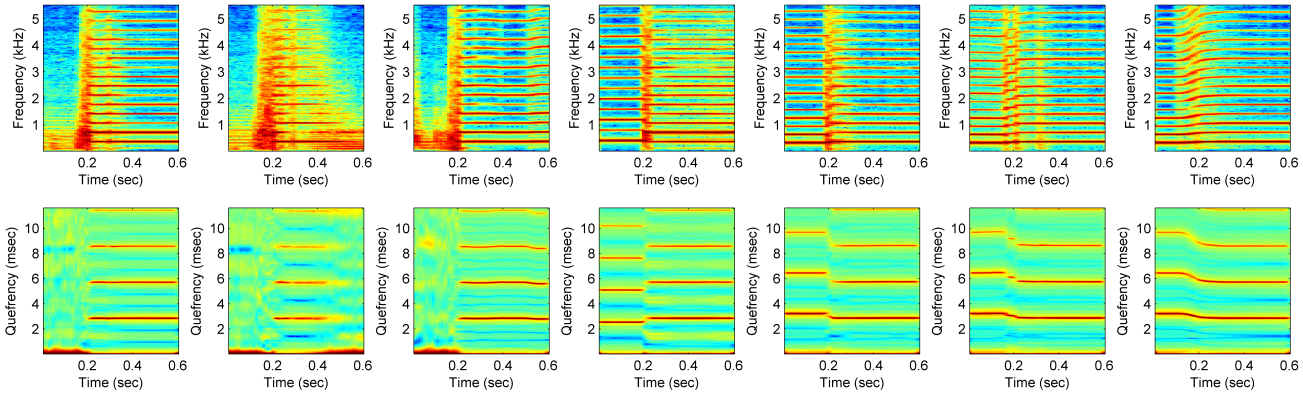

Figure 1. Spectrograms (the first row) and short-time cepstra (the second row) of the seven playing techniques considered in this study. From left to right: normal, muting, vibrato, pull-off, hammer-on, sliding, bending.

playing techniques are recorded with all the 6 strings. As a result, we can see from Table 1 that the numbers of clips of the 7 techniques are different, where normal has the largest number of 2,009 notes and muting has the smallest number of 385 notes. In total there are 6,580 clips.

5. METHODS

5.1 Feature representation

Our feature processing procedures have two steps: low-level feature extraction and sparse coding. In low-level feature extraction, we select spectrogram (SG), group-delay function (GDF), instantaneous frequency deviation (IFD), logarithm cepstrum (CL) and power cepstrum (CP), all of which are derived quantities from the short-time Fourier transformation (STFT):

$$\mathrm{S}^h(t,\omega) = \int x(\tau)\, h(\tau - t)\, e^{-j\omega\tau} d\tau = \mathrm{M}^h(t,\omega)\, e^{j\Phi^h(t,\omega)}, \quad (1)$$

where $x(t) \in \mathbb{R}$ is the input signal, $\mathrm{S}^h(t,\omega) \in \mathbb{C}$ stands for the two-dimensional STFT representation on time-frequency plane, and $h(t)$ refers to the window function. SG is the magnitude part of the STFT representation: $\mathrm{SG}^h(t,\omega) = |\mathrm{S}^h(t,\omega)|$. Phase spctrum is the imaginary part of the logarithm spectrum: $\Phi^h(t,\omega) = \mathrm{Im}\left(\log \mathrm{S}^h(t,\omega)\right)$. IFD and GDF are the derivative of phase Φ over time and frequency, respectively:

$$\mathrm{IFD}^h(t,\omega) = \frac{\partial \Phi^h}{\partial t} = \mathrm{Im}\left(\frac{\mathrm{S}^{\mathcal{D}h}(t,\omega)}{\mathrm{S}^h(t,\omega)}\right), \quad (2)$$

$$\mathrm{GDF}^h(t,\omega) = -\frac{\partial \Phi^h}{\partial \omega} - t = \mathrm{Re}\left(-\frac{\mathrm{S}^{\mathcal{T}h}(t,\omega)}{\mathrm{S}^h(t,\omega)}\right), \quad (3)$$

where $\mathcal{D}$ and $\mathcal{T}$ represent operators on window functions: $\mathcal{D}h(t) = h'(t)$ and $\mathcal{T}h(t) = t \cdot h(t)$. Detailed derivation procedures of GDF and IFD can be found in [2]. On the other hand, CL and CP are calculated as

$$\mathrm{CL}^h(t,q) = (\mathrm{S}^h)^{-1}\left(\log|\mathrm{S}^h(t,\omega)|\right), \quad (4)$$

$$\mathrm{CP}^h(t,q) = (\mathrm{S}^h)^{-1}\left(|\mathrm{S}^h(t,\omega)|^{1/3}\right), \quad (5)$$

where $(\mathrm{S}^h)^{-1}(\cdot)$ denotes the inverse STFT and q denotes quefrency [19]. Features derived from CL, such as the Mel-frequency cepstral coefficients (MFCCs), are often employed in audio signal processing [8, 16].

5.2 Sparse coding and dictionary learning

For any one of the aforementioned low-level features, denoted as $\mathbf{y} \in \mathbb{R}^m$, we further convert it to a sparse representation $\boldsymbol{\alpha} \in \mathbb{R}^k$ by SC. Specifically, SC involves the following l_1-regularized LASSO problem [7] to encode $\mathbf{y}$ over a given dictionary $\mathbf{D} \in \mathbb{R}^{m \times k}$.

$$\hat{\boldsymbol{\alpha}} = f_{\mathrm{SC}}(\mathbf{D},\mathbf{y}) = \arg\min_{\boldsymbol{\alpha}} \|\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1 . \quad (6)$$

The LASSO problem can be efficiently solved by for example the least angle regression (LARS) algorithm [7]. Moreover, the dictionary $\mathbf{D}$ is learned by the online dictionary learning (ODL) [15] implemented by the open-source package SPAMS (`http://spams-devel.gforge.inria.fr/`). The SC result when the input $\mathbf{y}$ is CL has been referred to as the sparse cepstral code [25].

6. EXPERIMENT

6.1 Experimental setup of individual notes

As Fig. 1 illustrates, the playing techniques can be better identified around the onsets for most cases. Therefore, our system starts from detecting the onset of each clip and then extracts features from each segment starting from the time before the onset by t_a second to the time after the onset by t_b second. We use the well-known spectral flux method [11] for onset detection, and empirically set $t_a = 0.1$ and $t_b = 0.2$ for all the clips. For STFT, we use Hanning window of window size 46 ms (1,024 samples) and hop size of 10 ms (441 samples). Under the sampling rate of 44.1 kHz, the dimension of all the low level features is 512 (i.e. considering only positive frequency).

We adopt a five-fold jack-knife cross-validation (CV) scheme for the evaluation. For all the fold partitions, the distribution of clips over the playing techniques is balanced. We learn both the classifier and the ODL dictionary from the training folds only, without using the test fold. The number of atoms k of each dictionary is set to

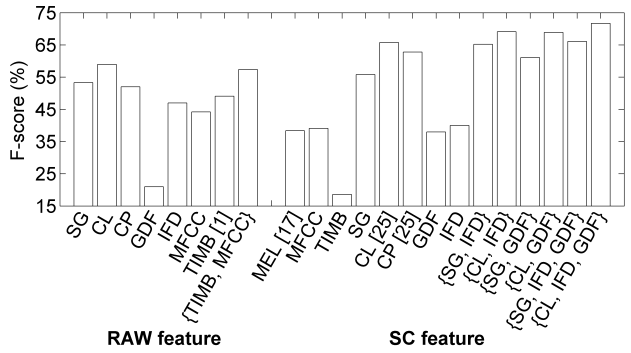

Figure 2. Average accuracies (in F-scores) of playing technique classification using various feature combination. Left part: RAW features; right part: SC features.

512. [3] After obtaining the frame-level sparse codeword $\boldsymbol{\alpha}$, a clip-level feature representation is constructed by mean pooling. Finally, the features, either with or without sparse coding, are fed into linear support vector machine (SVM) [9], with the parameter C optimized through an inside CV on the training data from the range $\{2^{-10}, 2^{-9}, \cdots, 2^{10}\}$. The evaluation results on the test set are reported in terms of F-score, which is the harmonic mean of precision and recall. All the evaluation is done at the clip-level.

We consider a number of baseline approaches for comparison. First, we use the MIRtoolbox (version 1.3.4) [14] to compute a total number of 41 features covering the temporal, spectral, cepstral and harmonic aspects of music signals (denoted as 'TIMB' in Fig. 2) as an implementation of a prior art on guitar playing technique classification [1]. Second, the conventional MFCC, ΔMFCC and $\Delta\Delta$MFCC are also used for their popularity (denoted as 'MFCC'). Third, we try the early fusion of MFCC and TIMB (i.e. by concatenating the corresponding clip-level representations to form a longer feature vector). Finally, for the features learned by SC, we note that the sparse representation of the mel-spectra (denoted as 'MEL') was used in [17], and the sparse representations of CL and CP were used in [25]. However, please note that the focus here is to compare the performance of using different features for the task, so our implementation does not faithfully follow the ones described in the prior arts. For example, Nam *et al.* uses automatic gain control as a pre-processing and uses multiple frame representation instead of frame-level features as input to feature encoding [17]. For simplicity the feature extraction and classification pipelines have been kept simple in this study.

We apply SC to all the five low-level features described in Section 5.1 and consider a number of early fusion of them. No normalization is performed for SC features. However, for non-SC features (referred to as 'RAW'), it is useful to apply a z-score normalization so that each feature dimension has zero mean and unit variance.

[3] Using an over-complete dictionary (i.e. $k \gg m$) usually improves the performance of SC features [25], but we leave this as a future work.

(a) SC+SG

	predicted class							F-score
	nor	mut	vib	pul	ham	sli	ben	
nor	**92.6**	3.26	1.07	1.01	0.85	0.90	0.38	55.2
mut	44.0	**43.7**	6.94	1.13	0.32	0.97	2.90	56.1
vib	31.0	4.93	**63.8**	0.27	0.00	0.00	0.00	74.1
pul	21.0	1.75	0.00	**21.8**	16.9	34.2	4.47	29.7
ham	31.4	0.36	0.18	12.6	**25.8**	25.6	4.14	33.1
sli	11.9	0.94	0.00	7.92	10.9	**52.7**	15.6	46.1
ben	3.56	0.92	0.11	2.18	1.26	14.5	**77.5**	75.6

(b) SC+CL

	predicted class							F-score
	nor	mut	vib	pul	ham	sli	ben	
nor	**95.6**	1.01	0.41	0.82	0.63	1.20	0.30	58.6
mut	38.9	**54.4**	4.35	0.16	0.00	0.65	1.45	66.3
vib	14.3	6.03	**79.7**	0.00	0.00	0.00	0.00	86.3
pul	27.2	0.58	0.19	**28.2**	14.6	25.6	3.69	38.9
ham	31.4	0.00	0.00	9.55	**38.2**	18.2	2.70	47.2
sli	14.9	1.42	0.00	4.43	7.26	**61.8**	10.2	56.7
ben	3.79	0.69	0.00	1.84	1.03	10.5	**82.2**	81.9

(c) SC+{CL,GDF,IFD}

	predicted class							F-score
	nor	mut	vib	pul	ham	sli	ben	
nor	**95.6**	1.59	0.33	0.55	0.79	0.79	0.36	64.1
mut	35.0	**57.9**	4.52	0.32	0.16	0.32	1.77	68.7
vib	12.3	6.85	**80.8**	0.00	0.00	0.00	0.00	86.9
pul	19.6	0.58	0.19	**41.2**	11.7	22.5	4.27	52.0
ham	24.3	0.18	0.00	10.5	**45.8**	17.5	1.80	55.2
sli	10.2	1.13	0.19	5.66	6.60	**70.4**	5.85	65.0
ben	1.38	0.23	0.00	0.23	0.80	5.17	**92.2**	89.4

Table 2. Confusion matrix (in %) of playing technique classification of electric guitar individual notes using different feature combinations.

6.2 Experiment results

From the left hand side of Figure 2, we find that both RAW+TIMB [1] and RAW+MFCC perform worse than RAW+SG, RAW+CL and RAW+CP, possibly because the feature dimension of the latter three is larger. However, after fusing TIMB and MFCC, the F-score is improved to 57.4%, which is not significantly worse than the result of RAW+CL (i.e. 59.0%) under the two-tailed t-test. It turns out that using sophisticated features such as those computed by the MIRtoolbox does not offer gain for this task. Note that the F-score of random guess would be 1/7=14.3%, because each fold is balanced across the 7 techniques. The performance of most RAW features is greatly better than the chance level.

In contrast, from the right hand side of Figure 2, we find that SC features usually performs much better than the non-SC (i.e. RAW) counterparts. For example, SC+SG, SC+CL and SC+CP are better than RAW+SG, RAW+CL and RAW+CP, respectively. These improvements are all significant under the two-tailed t-test ($p<0.01$, d.f.=8). Similar observations have been made in existing works that

apply SC features to MIR tasks (e.g. [17, 25]). We also find that using SC+CL already leads to significantly better F-score than RAW+{TIMB,MFCC} ($p<0.0001$, d.f.$=8$). Moreover, from the data of SC features we see that fusing GDF and IFD generally improves the accuracy, and that the best F-score 71.7% is obtained by fusing sparse-coded cepstral and phase features (i.e. SC+{CL,GDF,IFD}). The F-score of SC+{SG,GDF,IFD} is worse (66.1%) than SC+{CL,GDF,IFD}, but is still significantly better than SC+SG. We also note that SC does not improve the performance for MEL, MFCC, TIMB and IFD. This implies that sophisticated features like TIMB are not suitable for SC. Although SC+IFD is worse than IFD, its fusion with other SC features still results in better performance. In a nutshell, this evaluation shows that it is promising to use SC for playing technique classification, especially when we fuse multiple features derived from STFT.

Table 2 displays the confusion matrices for three different feature combinations with sparse coding. Table 2(a) shows the result of SC+SG, from which we see that normal and bending have relatively high F-scores of 74.1% and 75.6% (see the rightmost column), yet the other five techniques have F-scores lower. We see that many playing techniques can be easily misclassified as normal. We also see ambiguities between for example pull-off versus sliding and hammer-on versus sliding, showing that such techniques are difficult to be discriminated from one another in the logarithm-scale spectrogram.

In contrast, we see from Table 2(b) that SC+CL leads to consistent improvement in F-score for all the playing techniques, comparing to SC+SG. The largest performance gain (+14.1%) is obtained for hammer-on. We also see that the ambiguity between normal and vibrato is mitigated.

Finally, comparing Tables 2 (b) and (c) we see that SC+{CL,GDF,IFD} consistently improves the F-score for all the playing techniques. More interestingly, it seems that adding phase derivatives effectively alleviate the aforementioned confusions without compromising the discriminability of other classes. The F-scores of all the playing techniques are now above 50.0%.

7. REAL-WORLD MUSIC

The automatic transcription flow contains frame-level pitch detection, onset detection, and playing technique classification, one after another. We adopt the method proposed by Peeters [20] and use spectral and cepstral features for pitch detection. For onset detection, we use again the spectral flux method [4, 11]. Finally, we apply the playing technique classifier trained from the individual note dataset to classify the playing techniques of the guitar solo.

We present a qualitative evaluation of a real-world electric guitar solo excerpt performed by same professional guitarist. It is an interpretation of Sonata Artica's Tallulah released in 2001, for the fragment 3:59–4:08. We show in the first two subfigures of Fig. 3 its scoresheet and spectrogram. In the third subfigure we show the pitch and onset, using black horizontal bars, gray horizontal bars, and vertical dashed lines to denote the estimated frame-level pitches, ground truth pitches, and estimated onsets, respectively. We see that the estimated pitches and onsets match the ground truth quite well, except for some cases such as the mismatch between the onset at 7.70s and the change of pitch at 7.84s, which probably results from the ambiguity of the onset of bending.

The last subfigure of Fig. 3 compares the result of SC+SG and SC+{CL,GDF,IFD} for playing technique classification. Since our classification is performed with respect to the detected onsets, the errors in the stage of onset detection will fully propagate into the stage of playing technique classification. Therefore, the techniques which are not characterized by onset (e.g., a long-sustaining vibrato) cannot be transcribed. A true positive of onset is defined as an onset position which is detected within 100ms of the ground truth onset time. A true positive of playing technique is accordingly defined as a correct prediction of playing technique at a true positive of onset. We can see that the performance of playing classification degrades a lot in comparison to the case of individual notes. Specifically, we have 7 true positives (4 normal and 3 bending) for SC+{CL,GDF,IFD} and 5 true positives (2 sliding, 2 bending and 1 normal) for SC+SG, while there are in total 17 targets in the ground truth. The 2 muting at 2.38s and 4.60s and the hammer-on at 9.24 second are not recalled by both methods. Although SC+{CL,GDF,IFD} fails to recall sliding, SC+SG recalls 2 sliding. While SC+{CL,GDF,IFD} has many false positives of vibrato, SC+SG has many false positives of sliding. In general, SC+{CL,GDF,IFD} performs better.

The two estimated events at 4.11s and 5.80s are interesting. Although the two events do not present in the ground truth, the prediction of SC+{CL,GDF,IFD} is musically correct as the two false alarms of onset indeed occur in a long-sustaining vibrato. In contrast, SC+SG misclassifies the two events as pull-off and sliding, respectively.

8. CONCLUSION

In this study, we have reported a comparative study on the performance of a number of timbre modeling methods for the relatively unexplored task of guitar playing technique classification. The evaluation is performed on a large-scale individual-note dataset comprising of 6,580 clips and a real-world guitar solo recording. Our evaluation shows that sparse coding works well in learning features that are useful for the task, and that using features extracted from the cepstra and phase derivatives helps resolve the confusion among similar playing techniques. We also report a qualitative evaluation on guitar solo transcription. We are currently collecting more individual notes and solos to deeply understand the signal-level characteristics for these playing techniques. Although the present study might be at best preliminary, we hope it can call for more attention towards playing technique modeling.

9. ACKNOWLEDGMENTS

This work was supported by the Academia Sinica Career Development Award 102-CDA-M09.

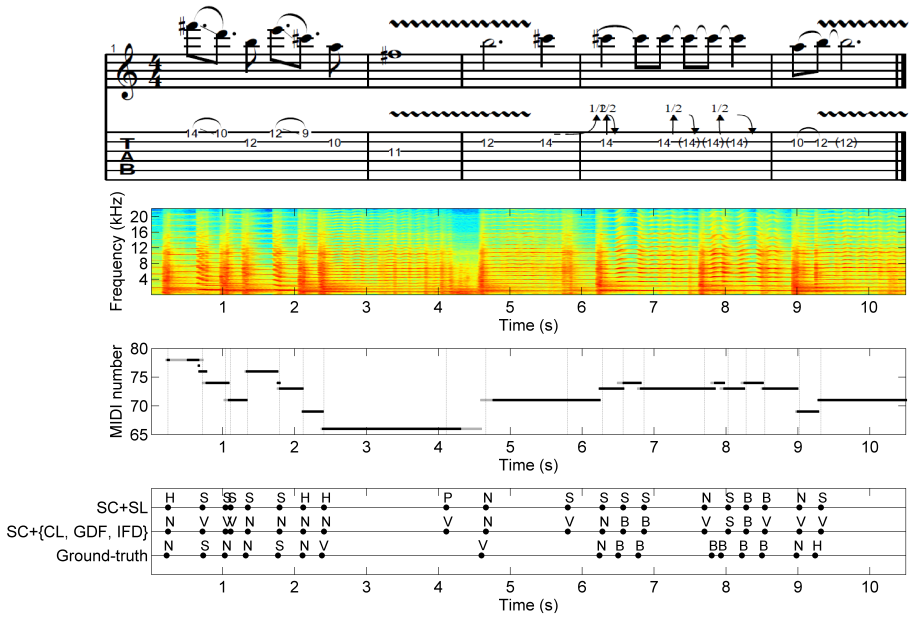

Figure 3. Result of transcribing a real-world guitar solo excerpt. From top to bottom: scoresheet, guitar tab, spectrogram, pitch and onset (gray bar: ground truth; black bar: estimated pitch; vertical dashed line: estimated onset), and result of playing technique classification by using SC+SG and SC+{CL,GDF,IFD}. Abbreviation: N=normal, V=vibrato, M=muting, P=pull-off, H=hammer-on, S=sliding, B=bending.

10. REFERENCES

[1] J. Abeßer et al. Feature-based extraction of plucking and expression styles of the electric bass guitar. In *ICASSP*, pages 2290–2293, 2010.

[2] F. Auger and P. Flandrin. Improving the readability of time-frequency and time-scale representations by the method of reassigment. *IEEE Trans. Sig. Proc.*, 43(5):1068–1089, 1995.

[3] A. M. Barbancho et al. Automatic transcription of guitar chords and fingering from audio. *IEEE Trans. Audio, Speech, and Language Processing*, 20(3):915–921, 2012.

[4] J. P. Bello et al. A tutorial on onset detection in music signals. *IEEE Speech Audio Process.*, 13(5-2):1035–1047, 2005.

[5] E. Benetos et al. Automatic music transcription: challenges and future directions. *J. Intelligent Information Systems*, 41(3):407–434, 2013.

[6] J. Dattorro. Effect design, part 2: Delay line modulation and chorus. *J. Audio engineering Society*, 45(10):764–788, 1997.

[7] B. Efron et al. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[8] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *ICASSP*, pages 753–756, 2000.

[9] R.-E. Fan et al. LIBLINEAR: A library for large linear classification. *J. Machine Learning Research*, 2008.

[10] P. Hamel et al. Automatic identification of instrument classes in polyphonic and pply-instrument audio. In *ISMIR*, 2009.

[11] A. Holzapfel et al. Three dimensions of pitched instrument onset detection. *IEEE Trans. Audio, Speech, Language Process.*, 18(6):1517–1527, 2010.

[12] E. J. Humphrey et al. Feature learning and deep architectures: new directions for music informatics. *J. Intelligent Information Systems*, 41(3):461–481, 2013.

[13] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*, chapter 6. Springer, 2006.

[14] O. Lartillot and P. Toiviainen. A Matlab toolbox for musical feature extraction from audio. In *DAFx*, 2007.

[15] J. Mairal et al. Online dictionary learning for sparse coding. In *Int. Conf. Machine Learning*, pages 689–696, 2009.

[16] M. Müller et al. Signal processing for music analysis. *IEEE J. Sel. Topics Signal Processing*, 5(6):1088–1110, 2011.

[17] J. Nam et al. Learning sparse feature representations for music annotation and retrieval. In *ISMIR*, pages 565–560, 2012.

[18] K. O'Hanlon and M. D Plumbley. Automatic music transcription using row weighted decompositions. In *ICASSP*, 2013.

[19] A. V. Oppenheim and R. W. Schafer. *Discrete-Time Signal Processing*. Prentice Hall, 2010.

[20] G. Peeters. Music pitch representation by periodicity measures based on combined temporal and spectral representations. In *ICASSP*, 2006.

[21] L. Reboursière et al. Left and right-hand guitar playing techniques detection. In *NIME*, 2012.

[22] L. Su and Y.-H. Yang. Sparse modeling for artist identification: Exploiting phase information and vocal separation. In *ISMIR*, pages 565–560, 2013.

[23] L. Su and Y.-H. Yang. Sparse modeling of subtle timbre: a case study on violin playing technique. In *WOCMAT*, 2013.

[24] K. Yazawa et al. Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency. In *ICASSP*, 2014.

[25] L.-F. Yu et al. Sparse cepstral codes and power scale for instrument identification. In *ICASSP*, 2014.