Sparse Modeling of Magnitude and Phase-Derived Spectra for Playing Technique Classification

Li Su, *Member, IEEE*, Hsin-Ming Lin, and Yi-Hsuan Yang, *Member, IEEE*

***Abstract*—Computational modeling of musical timbre is important for a variety of music information retrieval applications. While considerable progress has been made to recognize musical genres and instruments, relatively little attention has been paid to modeling playing techniques, which affect timbre in more subtle ways. In this paper, we contribute to this area of research by systematically evaluating various audio features and processing methods for multi-class playing technique classification, considering up to nine distinct playing techniques of bowed string instruments. Specifically, a collection of 6,759 chamber-recorded single notes of four bowed string instruments and a collection of 33 real-world solo violin recordings are used in the evaluation. Our evaluation shows that using sparse features extracted from the magnitude spectra and phase derivatives including group delay function (GDF) and instantaneous frequency deviation (IFD) leads to significantly better performance than using a combination of state-of-the-art temporal, spectral, cepstral and harmonic feature descriptors. For playing technique classification of violin singe notes, the former approach attains 0.915 macro-average F-score under a tenfold cross validation setting, while the latter only attains 0.835. Moreover, sparse modeling of magnitude and phase-derived spectra also performs well for single-note joint instrument-technique classification (F-score 0.770) and for playing technique classification of real-world violin solos (F-score 0.547). We find that phase information is particularly important in discriminating playing techniques with subtle differences, such as playing with different bowing positions (i.e., *normal*, *sul tasto*, and *sul ponticello*). A systematic investigation of the effect of parameters such as window sizes, hop factors, window types for phase-derived features is also reported to provide more insights.**

***Index Terms*—Group delay function, instantaneous frequency deviation, phase, playing technique classification, sparse coding.**

I. Introduction

MUSICAL instrument classification is important for music information retrieval (MIR) applications such as automatic transcription, source separation, music indexing and retrieval [1]. While early research tends to focus on monophonic signals or instrument solos, a considerable amount of effort has been made in recent years to deal with multiple-instrument music signals. Example topics of research include the identification of the predominant instrument in a polyphonic mixture, recognition of all the instruments, or estimation of the number of polyphony [2]–[8]. State-of-the-art techniques can extract instrument information from audio signals with a certain degree of accuracy [9]–[11]. We note that such effort addresses the *complexity* of musical timbre, with the intent of extracting timbre information from multi-source musical signals.

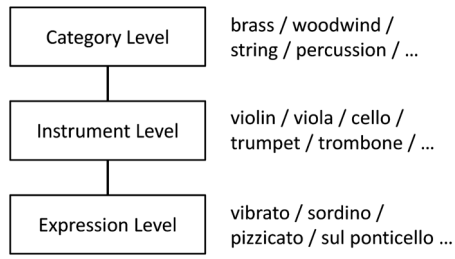


Fig. 1. Three subtlety levels of musical instrument timbre. Playing techniques belong to the expression level.

In addition to its complexity, music timbre can also be described in terms of its *subtlety*, which refers to the granularity of descriptors that differentiate between sound qualities. Fig. 1 illustrates the hierarchical architecture of instrument timbre according to subtlety. The top level of this figure is about the *instrument families*; the middle level considers each individual *instrument*; and the bottom *expression* level refers to all of the detailed audio effects within a particular instrument, such as various playing techniques associated with different instruments. Although playing techniques affect timbre in more subtle ways, it is a common practice for composers and musicians to employ different playing techniques to change the timbre of an instrument, either for aesthetic purposes or for expressing different emotions [12]–[14]. For example, *sordino* is a playing technique that decreases the volume of sound and gives a more mellow tone, with fewer audible overtones. Despite of that importance, relatively little attention has been paid to modeling timbre at the expression level so far.

To advance this area of research, we present in this paper a systematic evaluation of various state-of-the-art audio feature extraction and processing methods for the novel task of multi-class playing technique classification. As the first attempt, we start from chamber-recorded single notes of four bowed string instruments in this evaluation. Although one can argue that single notes are largely different from polyphonic signals, we opt for this simple setting to gain insight into the fundamental signal-level characteristics of different playing techniques and to lay the foundation of future studies. While the majority of the experiments are conducted over a large collection of 6,759 single notes (also referred to as individual or monophonic notes) sampled from the

Manuscript received May 01, 2014; revised August 24, 2014; accepted August 26, 2014. Date of publication October 08, 2014; date of current version October 16, 2014. This work was supported by an Academia Sinica Career Development Award. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Emmanuel Vincent.

L. Su and Y.-H. Yang are with the Research Center for Information Technology Innovation, Academia Sinica, Taipei 11564, Taiwan (e-mail: lisu@citi.sinica.edu.tw; yang@citi.sinica.edu.tw).

H.-M. Lin is with the Music Department, University of California at San Diego, La Jolla, CA 92093 USA.

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.2014.2362006

well-known RWC database [15], we also report a preliminary experiment evaluating playing technique classification using a small collection of 33 real-world violin solo fragments collected from a music textbook and online resources. We consider up to nine distinct playing techniques of bowed string instruments, with an in-depth discussion of the main findings of the evaluation.

Many musical timbre features and relevant feature processing methods have been proposed in the MIR literature [1]. In this study, we consider a number of temporal, spectral, cepstral and harmonic features computed from the time-domain waveforms and the magnitude spectra, such as Mel-frequency cepstral coefficients (MFCC) and linear predictive coding (LPC) coefficients [3]. Moreover, we explore features extracted from the phase-derived terms such as group delay function (GDF) and instantaneous frequency deviation (IFD) [16]–[18], since we presume that phase information might be important in discriminating subtle differences in musical timbre.

Before using the features as input to train a classifier, we also experiment with popular feature processing methods including feature selection, dimension reduction [4], vector quantization (VQ) and sparse coding (SC) [6]–[8] for better accuracy. Our main findings include:

- evaluation on both the single-note dataset and the violin solo dataset shows that the best accuracy of playing technique classification is obtained by sparse modeling of magnitude and phase-derived spectra. Conventional timbre features also perform well but are less effective.
- phase information nicely complements information from the magnitude part of the spectrogram, leading to pronounced improvement in modeling challenging cases such as *flageolet*, *non-vibrato* and *sul ponticello*.
- for single notes, we are able to discriminate playing techniques of different bowed string instruments in a joint instrument-technique classification experiment.
- performance degrades when we consider real-world violin solos but the performance is still promising.

The rest of this paper is organized as follows. Section II introduces the playing techniques considered in this study and presents a signal-level analysis of the techniques. Section III provides an overview of the study. Section IV describes the audio features and feature processing methods, including details about the proposed sparse modeling of magnitude and phase-derived spectra approach. Section V describes the result of the performance study, with discussions in Section VI. Finally, we conclude the paper in Section VII.

II. Signal Analysis of Instrument Sounds

Classical music theorists and composers have adopted unusual playing techniques, leading to various timbres of bowed string instruments.[1] As early as 1542, Sylvestro di Ganassi suggested that playing near the bridge produces a harsh sound [12], which is the modern *sul ponticello* technique. Joseph Haydn also used this effect, starting from measure 84 in the second movement of his Symphony No. 97 (1792) [13]. In this study, we consider the nine playing techniques of bowed string instruments described in Table I. These techniques include various excitation mechanism (bowing/fingering), input location (near the bridge/fingerboard) and time-varying dynamics (vibrato/non-vibrato). An example musical score of a real-world string quartet containing different playing techniques is shown in Fig. 2(a). More information about the techniques can be found in music textbooks (e.g. [14]).

TABLE I
Nine Playing Techniques of Bowed String Instruments

Technique	Description
Flageolet	Touching a string lightly at various points called nodes with the left hand and playing the bow with the right hand to produce either artificial or natural harmonics.
Normal	Pressing the finger firmly on the string at the desired pitch while quickly rocking it back and forth on the string. This technique is also referred to as *vibrato*, but in the RWC database [15] it is labeled as *normal*, possibly because it is the most common playing technique.
Non-vibrato	Pressing the finger firmly on the string at the desired pitch but without rocking it, so as to produce pale sound.
Pizzicato	Plucking the strings by fingers.
Sordino	Placing a small plastic, wooden, or metal object on the bridge (cf. Fig. 2) to absorb some of the vibrations and thereby create a softer and smoother sound.
Spiccato	An effort to make the bow bounce. The RWC database [15] only contains samples of 'conscious' spiccato, although there are also 'spontaneous' or 'slurred' spiccato.
Sul ponticello	Playing on or near the bridge to produce upper partials of a tone that are not normally heard; see Fig. 2.
Sul tasto	Playing on or near the fingerboard to produce a flutelike, soft and hazy tone; see Fig. 2.
Tremolo	Repeating a pitch as often as possible during the length of the written note by means of short, quick up-and-down bow or finger strokes. We consider only bowed tremolo (instead of fingered tremolo) in this study.

As a preliminary signal-level analysis of the playing techniques, Fig. 3 shows the magnitude spectra and phase profiles of violin A4 single notes played by five different techniques. The single notes are one-second fragments clipped from the onset time and sampled at 44,100 Hz. We set the window size of short-time Fourier transform (STFT) to 2,048 samples, hop factor (i.e. percentage of overlap between consecutive windows) to 10% and frequency resolution (after zero-padding) to 1 Hz. As the first row of Fig. 3 shows, although the magnitude spectra of the playing techniques resemble one another, especially for the first few harmonics, there are still some visually discernible differences. For example, *sordino* and *sul tasto* have much weaker overtones (i.e. higher harmonic components) than the other three techniques; *sul ponticello* generates more energy in high frequencies and contains a lot of noise-like terms. The techniques *normal*, *non-vibrato* and *sul ponticello* have relatively stronger 2nd and 5th harmonics and weaker 1st, 3rd and 4th harmonics, while the 5th harmonic is suppressed (by using a muting device) in *sordino*. This suggests that it is possible to use features computed from the magnitude spectra to classify the playing techniques.

On the other hand, as the last two rows of Fig. 3 show, from the phase profiles one can also mine patterns potentially useful for playing technique classification. For example, from the phase spectra (i.e. second row) we see that the phase of *sul ponticello* features irregular fluctuations over the entire spectrum; *non-vibrato* and *sul tasto* fluctuate only in high frequencies; in contrast, *normal* and *sordino* have less noisy parts, and most

[1]Videos by Barnes and Mullins that distinguish between those timbres are accessible online at http://www.youtube.com/user/BarnesandMullinsUK/

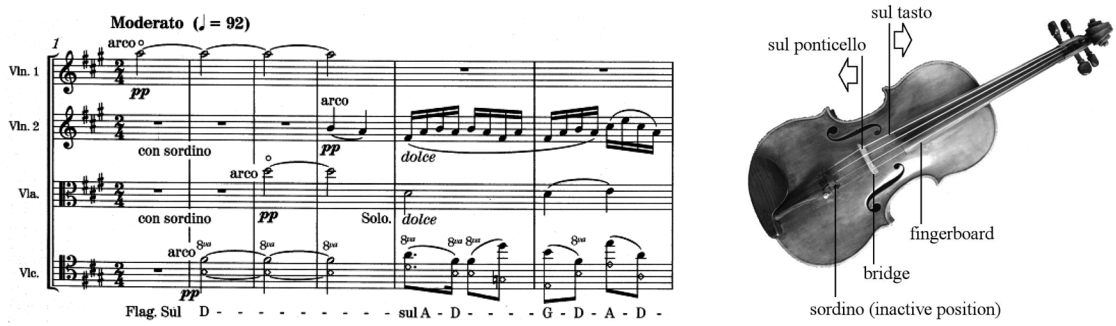


Fig. 2. (a) Musical score of String Quartet No. 1, mov. 3, Trio, mm. 1-6, by Alexander Borodin. It contains the playing techniques *normal* (i.e. *vibrato*), *sordino* (labeled as "con sordino") and *flageolet* (labeled as "Flag."). (b) A photo of viola, with indication of fingerboard, bridge, the position placing the mute (for sordino) and the bowing position for *sul ponticello* and *sul tasto*. Photographed by Chin-Ting Huang.

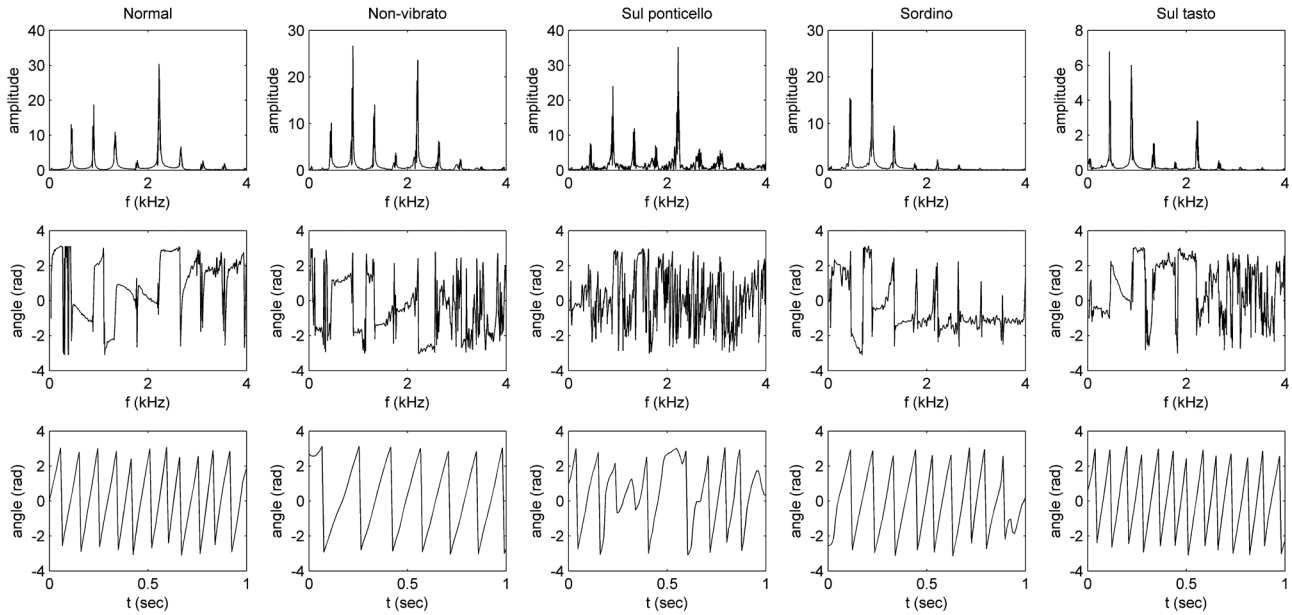


Fig. 3. Magnitude spectra and phase profiles of violin A4 tones played using the following five different techniques: (from left to right) *normal*, *non-vibrato*, *sul ponticello*, *sordino* and *sul tasto*. The displayed signals are: (from top to bottom) the magnitude spectrum of the time frame 0.5 second after the onset time, the corresponding phase spectrum, and the signal phase with respect to time at the frequency bin closest to the fundamental frequency. Please note that the scale of the magnitude spectra is not consistent because *sul tasto* is only played at the dynamic level of *piano* in the RWC database (cf. Section III).

of the harmonic spurs are identifiable from the phase spectra. From the change of phase with respect to time (i.e. third row) we see *non-vibrato* has relatively smaller phase slope, whereas the pattern is much irregular for *sul ponticello*. Motivated by the above observations, we also explore phase-related features in this study.[2]

We note that although phase information has contributed to the signal processing of speech and music,[3] audio features used in existing music classification systems are mostly computed from the magnitude and harmonic parts of the spectra [27], leaving the phase part less exploited. One exception is the work presented by Su and Yang [28], which uses GDF to model the singing voice of pop music artists. However, to our best knowledge, the performances of different magnitude- and phase-derived features have not been compared systematically in the context of musical timbre modeling.

Phase information has been relatively less explored possibly due to its dependence on the analysis time frame and window function [20], its sensitivity to noises, and the limitation of phase wrapping algorithms. These issues present a challenge in reliably estimating phase for real-world signals. Moreover, in discrete-time processing, many of the transmission zeros of the STFT operator are close to the unit circle in the z-plane, leading to unstable spikes when computing the derivatives of the phase spectrum (or *phase derivatives*) such as GDF and IFD [29]. Although there are advanced methods to compute more reliable phase features [21][23], our study provides empirical evidences (cf. Section V) showing that useful patterns can be mined from the fundamental forms of the phase derivatives, given that they are further processed by sparse coding techniques [30]–[33].

[2]The phase slopes shown in the third row of Fig. 2 are indicative of the IFD, which is the derivative of phase angle over time (cf. Section IV-A). Please note that they are not the instantaneous frequencies, because the time frames are not sampled fast enough to avoid aliasing.

[3]For example, pitch shifting [19], speech synthesis [20], speaker recognition [21], [22], [23], melody tracking [24] and musical onset detection [25], [26].

TABLE II
NUMBER OF SAMPLES FOR EACH PLAYING TECHNIQUE IN THE SINGLE-NOTE RWC DATASET AND THE REAL-WORLD DATASET

Playing technique	Single note from RWC				Real-world
	violin	viola	cello	bass	violin
flageolet	36	87	54	–	–
normal	386	361	378	362	7
non-vibrato	129	122	125	61	–
pizzicato	375	349	356	184	8
sordino	123	126	129	–	–
spiccato	379	365	373	–	7
sul ponticello	130	126	129	66	7
sul tasto	99	114	130	–	4
tremolo	360	361	384	–	–
total	2,017	2,011	2,058	673	33

III. OVERVIEW OF THE STUDY

Two datasets are used. The first one is composed of chamber-recorded single notes of four bowed string instruments (violin, cello, viola and contrabass) collected from the RWC musical instrument sound database [15]. The RWC dataset usually contains samples of all possible pitch ranges, three dynamic levels (i.e. *forte*, *mezzo forte* and *piano*) and three brands of instruments for each playing techniques, but there are some exceptions. For example, *sul ponticello* is played only in *forte* (f); *non-vibrato*, *flageolet* and *sordino* are played only in *mezzo forte* (mf), and the recordings for *sul tasto* are played only at the *piano* (p) dynamic level. Moreover, as shown in Table II, only four playing techniques are available for contrabass (denoted as 'bass' in the table). In total, 6,759 sound samples are taken from the RWC database. All notes are sampled at 22,050 Hz and clipped into segments by the root-mean-square signal energy. The clips cover the attack, decay, sustain and release (ADSR) parts of the entire signal. On average, the length of each clip is 2.8 seconds.

The second dataset is composed of real-world music clips. This dataset contains 33 solo violin recordings, among which twelve are gathered from the audio examples in a music textbook [14] and the others from YouTube. Each recording is clipped such that each contains a sequence of notes played with only one playing technique. The dataset contains solos of five playing techniques (see Table II), and the length of each clip ranges from 5 to 15 seconds. Detailed metadata about the clips are available online.[4]

We consider three classification settings: 1) nine-class playing technique classification of violin singe notes across the nine techniques using the RWC dataset; 2) sixteen-class joint instrument-technique classification across the four instruments and the four respective common techniques using the RWC dataset; and 3) five-class playing technique classification of violin solos using the real-world dataset.

As described in the next section, we experiment with various audio features and feature processing methods in this study. The support vector machine (SVM) [34][35] is used for classifier training and prediction for its superior performance in existing studies on music classification [36]. We will also perform a parameter sensitivity test to investigate how the performance is influenced by changes in some parameters.

[4]http://mac.citi.sinica.edu.tw/violin-playing-technique/

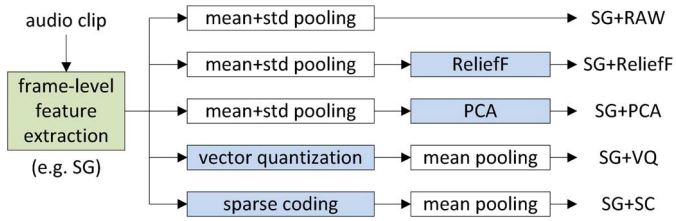


Fig. 4. Flow diagram of frame-level feature extraction, clip-level temporal aggregation (pooling) and frame-level or clip-level feature processing.

IV. FEATURE EXTRACTION AND CLASSIFICATION

Fig. 4 illustrates the flow diagram of feature representation computation. It involves the following three procedures: *frame-level feature extraction* converts an input audio clip to a number of frame-level feature vectors; *clip-level temporal aggregation* pools the frame-level feature vectors into a clip-level feature vector for the audio clip; and *feature processing* processes the (resp. frame-level or clip-level) feature vectors either before or after pooling.[5] The final clip-level feature vectors are used for classifier training and testing.

A. Frame-level Feature Extraction

The following features are considered:

- **TIMBRE** is a rich set of temporal, spectral, cepstral and harmonic descriptors computed by the MIRtoolbox (version 1.3.4) [27]. It contains spectral centroid, brightness, spread, skewness, kurtosis, roll-off, entropy, irregularity, roughness, inharmonicity, flux, zero-crossing rate, low energy ratio, 10-order LPC coefficients, 40-D bark band energy, 40-D Mel-spectrum, 40-D MFCC, and the first-order time difference for all the above features, totalling 290 features. Such a hybrid set of features is commonly employed in music classification problems [36]. We note that none of them directly contain phase information.
- **MFCC**, as a subset of TIMBRE, is composed of the 40-D MFCC and its first-order time difference (i.e. Δ MFCC). It is a standard feature set in audio processing [36].
- **Time-frequency features** include the log-scaled spectrum (**SG**), unwrapped phase (**PH**), group delay function (**GDF**) and instantaneous frequency deviation (**IFD**). Moreover, we consider the pairwise fusion (i.e. {SG, GDF}, {SG, IFD} and {GDF, IFD}) and the triple fusion (i.e. {SG, GDF, IFD}; also denoted as **ALL**) of the last three features. The fusion is performed by concatenating the corresponding clip-level representation of each feature (i.e. early fusion). In our implementation, the extraction of these time-frequency features is based on the time-frequency toolbox [38].

We describe the time-frequency features in more details below. Consider a general representation of STFT of an input signal $x(t) \in \mathbb{R}$:

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

[5] We follow the common practice in the literature to employ SC and VQ at the frame-level [6], [7], [8] and RAW, PCA and ReliefF at the clip level [9], [37]. We have experimented with the reverse setting (e.g. performing SC at the clip-level) and found that it is better to follow the convention.

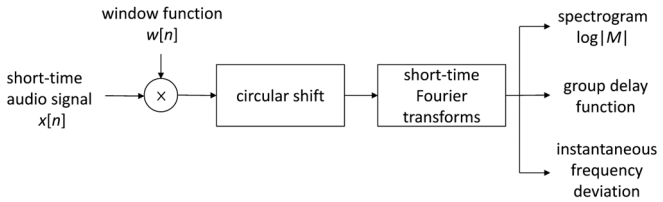


Fig. 5. The process of extracting the magnitude and phase-derived spectra from audio signals.

$$= \mathrm{M}_x^h(t,\omega)\, e^{j\Phi_x^h(t,\omega)}, \tag{2}$$

where $\mathrm{S}_x^h(t,\omega) \in \mathbb{C}$ stands for the two-dimensional STFT representation on time-frequency plane and $h(t)$ refers to the window function. Representing the amplitude $\mathrm{M}_x^h(t,\omega)$ and phase $\Phi_x^h(t,\omega)$ of (1) on the time-frequency plane allows us to obtain (2). By taking the natural logarithm of Eq. (2), the real and imaginary parts of (2) correspond to SG, $\log\|\mathrm{M}_x^h(t,\omega)\| \in \mathbb{R}$ and PH, $\Phi_x^h \in \mathbb{R}$, respectively:

$$\log\|\mathrm{M}_x^h(t,\omega)\| = \mathrm{Re}\left(\log \mathrm{S}_x^h(t,\omega)\right) \tag{3}$$

$$\Phi_x^h(t,\omega) = \mathrm{Im}\left(\log \mathrm{S}_x^h(t,\omega)\right). \tag{4}$$

IFD is defined as the derivative of phase (4) over time:

$$\mathrm{IFD}_x^h(t,\omega) = \frac{\partial \Phi_x^h(t,\omega)}{\partial t} = \mathrm{Im}\left(\frac{\mathrm{S}_x^{\mathcal{D}h}(t,\omega)}{\mathrm{S}_x^h(t,\omega)}\right). \tag{5}$$

GDF is the negative derivative of phase (4) over frequency:[6]

$$\mathrm{GDF}_x^h(t,\omega) = -\frac{\partial \Phi_x^h(t,\omega)}{\partial \omega} - t = \mathrm{Re}\left(-\frac{\mathrm{S}_x^{\mathcal{T}h}(t,\omega)}{\mathrm{S}_x^h(t,\omega)}\right). \tag{6}$$

$\mathcal{D}$ and $\mathcal{T}$ represent operators on window functions such that

$$\begin{aligned} \mathcal{D}h(t) &= h'(t), \\ \mathcal{T}h(t) &= t \cdot h(t). \end{aligned} \tag{7}$$

Detailed derivation procedures of GDF and IFD can be found in related work on time-frequency reassignment [16][17]. Equations (5) to (7) are used to compute the phase derivatives throughout the study. See Appendix 7 for more discussion on the physical meanings of GDF and IFD.

Fig. 5 illustrates the procedure of extracting the time-frequency feature. We first segment the signal into short-time frames by multiplying a series of windows with length T and a hop factor H. The window function contributes to the additional phase delay, which can be compensated by circularly shifting the signal by $T/2$ before performing STFT [22]. Unless otherwise specified, we set the window size T to 2,048 and the hop factor H to 10% in this study.

B. Feature Processing

We consider five feature processing methods:

- **RAW**: no processing.
- **Feature selection by ReliefF**: evaluates the importance of each feature and discards the less important ones in a supervised fashion. It measures feature importance from the training data by comparing the distance between the nearest neighbor from the same class (nearest hit) and to the nearest neighbor from a different class (nearest miss) according to the feature [39]. If the average distance to the nearest hit is shorter, the feature would be considered more important. We select the top-128 important features by the ReliefF routine of the MATLAB statistics toolbox [40] and discard the remaining features.
- **Principal component analysis (PCA)**: projects the feature space to another space constructed by its principal axes using a linear transformation matrix. For fair comparison with ReliefF, we also reduce the number of features to 128 after performing PCA. Both ReliefF and PCA are performed at the clip level (i.e. after pooling).
- **Vector quantization (VQ)**: For VQ and SC, we convert a frame-level feature vector $\mathbf{x} \in \mathbb{R}^m$ into the encoding result $\alpha \in \mathbb{R}^k$ using a *dictionary* (i.e. audio codebook) $\mathbf{D} \in \mathbb{R}^{m\times k}$, where k denotes the number of codewords in $\mathbf{D}$. When there are t input vectors from an audio clip, we encode each vector individually. VQ is a classic feature encoding method. It encodes the features by searching for the codeword in $\mathbf{D}$ that is the closest to the input $\mathbf{x}$ in terms of Euclidean distance [41]. The method can be viewed as a special case of an l_0-regularized least-square error problem with the constraint $\|\boldsymbol{\alpha}\|_0 = 1$ (number of non-zero terms being only one):

$$\hat{\boldsymbol{\alpha}} = \arg\min_{\hat{\boldsymbol{\alpha}}} \|\mathbf{x} - \mathbf{D}\hat{\boldsymbol{\alpha}}\|_2^2, \text{ subject to } \|\boldsymbol{\alpha}\|_0 = 1. \tag{8}$$

- **Sparse coding (SC)** represents the input signal $\mathbf{x}$ by a sparse combination of the dictionary codewords by solving the following LASSO problem [30],

$$\boldsymbol{\alpha}^* = \arg\min_{\boldsymbol{\alpha}} \quad \frac{1}{2}\|\mathbf{x} - \mathbf{D}\boldsymbol{\alpha}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_1, \tag{9}$$

where λ controls the balance between the reconstruction error $\|\mathbf{x} - \mathbf{D}\alpha\|_2^2$ and the sparsity $\|\alpha\|_1 = \sum|\alpha_j|$, which is a convex relaxation of $\|\alpha\|_0$. The encoding result α is therefore the weighting of the codewords representing the input feature $\mathbf{x}$. The problem can be solved efficiently by off-the-shelf programs such as the LARS-LASSO [31]. According to the classical normalization factor [33], we set λ to $m^{-1/2}$. Both VQ and SC are performed at the frame level (i.e. before pooling).

Robust signal reconstruction approaches including sparse coding and dictionary learning have received considerable attention in recent years and their variants have been used extensively in various audio-related research [42]–[46]. These algorithms aim at constructing a succinct representation of raw features as a combination of only a few codewords from the dictionary [33], [47]. Such a sparse representation can accurately capture the essential components of a signal while discarding irrelevant information such as outliers and noises. When $k > m$, it has been shown that SC outperforms VQ for audio classification problems [43]. It has also been found that SC works better with primitive features (e.g. SG, GDF and IFD) that preserve more details of the raw signal [43].

[6] Taking the derivative of Eq. (4) with respect to frequency produces a time delay term t, which is discarded as we are more interested in the group delay within the frame instead of the delay from the global time reference.

The dictionary $\mathbf{D}$ used in both VQ and SC is constructed from an external audio collection using an l_1-regularized online dictionary learning (ODL) algorithm [33] in an unsupervised fashion. The audio collection used for dictionary learning, referred to as the *training corpus*, has no overlap with the dataset for classifier training and testing. Specifically, as the RWC database [15] contains three brands for each instrument, we use one brand as the training corpus and the other two for classifier training and testing. The dictionaries are constructed from each instrument separately, and the dictionary size k is set to 1,024. Although previous works reveals that a larger dictionary generally yields better performance, that would incur heavy computational costs [44].

We refer readers to Appendix B for the algorithmic details of ODL. We note that it has been found [47] that learning a dictionary from data using algorithms such as k-means clustering or ODL works better than using an analytic dictionary constructed, for example, using Gammatone filters [7] or Gabor wavelets [8]. Building a *random* dictionary by either randomly extracting exemplars from the training corpus [44] or using a randomized clustering forest [45] has also been found efficient and effective. However, for simplicity we opt for using ODL for dictionary learning and do not consider other alternatives. We use the sparse modeling software (SPAMS) [33] for implementing the LASSO solver and ODL.

C. Clip-Level Temporal Aggregation and Post-processing

For features processed by RAW, ReliefF or PCA, we take the mean and standard deviation across time for temporal aggregation (i.e. $\mathrm{mean} + \mathrm{std}$ pooling), following [9]. In this way, the length of a clip-level feature vector for $\mathrm{MFCC} + \mathrm{RAW}$ and $\mathrm{TIMBRE} + \mathrm{RAW}$ would be 160 and 580, respectively. In addition, as a post-processing, for features processed by RAW, ReliefF or PCA, we further normalize the resulting clip-level features by taking the z-scores, subtracting each feature by its mean and then dividing by its standard deviation [36], before using them in the classification stage.

For features processed by VQ and SC, we employ *mean pooling* that takes the average of the frame-level encoding results α_i within a clip over time to form the clip-level feature $\beta = \frac{1}{n}\sum_{i=1}^{n}\alpha_i$. Each component of β represents the average *term occurrence* of the dictionary codewords in the audio clip [43]. Moreover, as a post-processing, we perform square-root power normalization on β according to $\mathrm{sgn}(\beta)|\beta|^{0.5}$, where $\mathrm{sgn}(\cdot)$ denotes the sign function. Power normalization has been shown effective in suppressing anomalies such as non-additivity and non-normality in the data [43]–[48].

D. Classification

For features processed by RAW, ReliefF and PCA, we use radial-basis function (RBF) kernel SVM implemented by LIBSVM [34], fixing the SVM parameters C and γ to 2^4 and 2^{-8} empirically after a pilot study. For features processed by VQ and SC, we use linear kernel SVM implemented by LIBLINEAR [35], as linear SVM has been shown effective for high-dimensional but sparse features such as the output of VQ and SC [43][44]. We fix C to 2^4 for the linear SVM.

TABLE III
ACCURACIES (IN MACRO-AVERAGE F-SCORES) OF PLAYING TECHNIQUE CLASSIFICATION OF VIOLIN SINGLE NOTES COLLECTED FROM RWC: DIFFERENT FRAME-LEVEL FEATURES AND PROCESSING METHODS

Frame-level feature	Feature processing method				
	RAW	ReliefF	PCA	VQ	SC
MFCC	0.654	0.680	0.727	0.485	0.630
TIMBRE	0.743	0.821	**0.835**	0.477	0.351
PH	0.149	0.349	0.438	0.373	0.429
SG	0.442	**0.788**	0.781	0.390	0.809
GDF	0.113	0.444	0.402	0.363	0.587
IFD	0.112	0.749	0.493	0.317	0.738
{SG, GDF}	0.112	0.788	0.607	0.520	**0.870**
{SG, IFD}	0.118	0.788	0.595	0.470	**0.903**
{GDF, IFD}	0.113	0.749	0.551	0.451	0.807
ALL	0.118	0.788	0.534	0.575	**0.915**

V. PERFORMANCE EVALUATION

As described in Section III, we consider three classification settings in this evaluation: playing technique classification of violin single notes, joint instrument-technique classification of single notes, and an evaluation on real-world violin solos.

A. Playing Technique Classification of Violin Single Notes

The data are divided into ten folds, by using nine folds for training and one fold for testing at a time in an (outer) cross-validation (CV) setting. We repeat the random partitioning ten times and report the average result for the 100 runs (i.e. ten-fold results for ten fold partitions). Macro-average F-score is used as the evaluation metric:

$$F = \frac{1}{d}\sum_{i=1}^{d}\frac{2P^{(i)}R^{(i)}}{P^{(i)}+R^{(i)}}, \tag{10}$$

where d is the number of classes, $P^{(i)}$ is the precision rate $N_{tp}^{(i)}/(N_{tp}^{(i)}+N_{fp}^{(i)})$, $R^{(i)}$ is the recall rate $N_{tp}^{(i)}/(N_{tp}^{(i)}+N_{fn}^{(i)})$, and $N_{tp}^{(i)}$, $N_{fp}^{(i)}$, $N_{fn}^{(i)}$ is the number of true positives, false positives and false negatives for the i-th class, respectively.

Table III summarizes the results of different combinations of features and processing methods.

- From the first three columns of Table III, we see that TIMBRE and MFCC (i.e. rows 1–2) perform better than the time-frequency features (rows 3–6) when we process the features by RAW, ReliefF or PCA. The combination $\mathrm{TIMBRE} + \mathrm{PCA}$ leads to the best average F-score 0.835, showing that the hybrid feature set computed by the MIRtoolbox [27] is effective for the task.[7] As for the time-frequency features, we see that processing the features by ReliefF and PCA is helpful, but even for the best combination $\mathrm{SG} + \mathrm{ReliefF}$ the F-score reaches only 0.788. Moreover, the F-scores of the phase-related features generally fall below 0.500. Fusing the time-frequency features (i.e. rows 7–10) results in no gain.
- From the last two columns of Table III, however, we see that the time-frequency features perform much better than RAW, ReliefF or PCA when being processed by SC. In addition, SC performs much better than VQ, possibly owing to its robustness to noise and the soft- rather than

[7] Although not shown in Table III, reducing the number of features to values other than 128 by ReliefF or PCA does not further improve the accuracy much.

TABLE IV
CONFUSION MATRIX (IN %) OF PLAYING TECHNIQUE CLASSIFICATION OF VIOLIN SINGLE NOTES FROM RWC USING DIFFERENT FEATURES (a) TIMBRE + PCA, (b) SG + SC, (c) ALL + SC

(a)

		predicted class								
		flag	norm	novi	pizz	sord	spic	supc	suta	trem
actual class	flageolet	**58.2**	3.59	23.3	0.00	2.31	0.00	4.36	7.44	0.77
	normal	0.09	**91.8**	3.15	0.09	0.00	0.68	0.02	3.68	0.47
	non-vibrato	3.41	12.7	**78.4**	0.07	0.44	0.67	1.48	2.44	0.30
	pizzicato	0.00	0.42	0.26	**98.4**	0.31	0.50	0.00	0.10	0.05
	sordino	0.00	3.23	0.37	1.52	**93.5**	0.85	0.00	0.49	0.06
	spiccato	0.00	0.62	0.00	1.34	0.77	**96.2**	0.18	0.23	0.70
	sul ponticello	0.50	6.26	4.03	0.07	2.07	1.44	**83.2**	2.37	2.01
	sul tasto	0.24	26.2	2.80	0.08	0.72	0.80	4.88	**63.9**	0.32
	tremolo	0.00	1.65	0.00	0.78	0.23	1.00	0.15	0.90	**95.3**

(b)

		predicted class								
		flag	norm	novi	pizz	sord	spic	supc	suta	trem
actual class	flageolet	**55.7**	17.4	5.90	0.00	0.77	2.56	1.79	7.95	7.95
	normal	0.26	**85.7**	3.12	0.00	0.91	4.45	1.24	2.12	2.19
	non-vibrato	2.44	29.3	**50.1**	0.00	3.26	6.52	0.44	5.33	2.59
	pizzicato	0.00	0.00	0.00	**98.5**	0.00	1.20	0.00	0.00	0.26
	sordino	0.00	2.62	2.26	0.00	**92.3**	2.80	0.00	0.00	0.00
	spiccato	0.00	1.68	0.34	1.57	0.18	**95.6**	0.26	0.15	0.18
	sul ponticello	0.58	6.55	1.73	0.00	0.00	2.23	**79.7**	0.22	8.99
	sul tasto	1.04	23.6	3.68	0.00	0.24	1.12	0.08	**66.5**	3.76
	tremolo	0.00	1.73	2.08	0.25	0.00	1.48	1.73	1.68	**91.1**

(c)

		predicted class								
		flag	norm	novi	pizz	sord	spic	supc	suta	trem
actual class	flageolet	**77.0**	4.62	3.85	0.00	0.00	0.00	2.82	8.21	3.59
	normal	0.00	**94.5**	1.40	0.00	0.68	1.72	0.56	0.79	0.40
	non-vibrato	0.52	17.2	**74.4**	0.74	0.59	0.22	0.15	6.15	0.07
	pizzicato	0.00	0.00	0.00	**98.1**	0.00	1.60	0.00	0.00	0.26
	sordino	0.00	0.55	0.06	0.61	**97.3**	1.52	0.00	0.00	0.00
	spiccato	0.03	0.08	0.26	1.13	0.00	**98.3**	0.00	0.00	0.26
	sul ponticello	0.00	1.01	0.50	0.29	0.00	0.00	**95.9**	0.00	2.30
	sul tasto	0.24	17.3	3.76	0.00	0.24	1.36	0.00	**75.3**	1.84
	tremolo	0.03	0.48	0.33	0.38	0.00	0.50	0.00	0.03	**98.3**

hard-assignment strategy [30], [31]. SG + SC performs the best among the non-fused time-frequency features (i.e. rows 3–6). We also see that the phase-derivatives GDF and IFD perform better than the raw phase spectra PH. Fusing the sparse representation of magnitude and phase-derived spectra further improves the accuracy. For example, SG, GDF + SC attains F-score 0.870, which is significantly (p-value <0.05, d.f. $= 198$) better than the result of TIMBRE + PCA under the two-tailed t-test.[8] The triple fusion ALL + SC obtains the best F-score 0.915 among all the considered compilations.

Table IV displays the confusion matrices for three different feature combinations. From the result of TIMBRE + PCA (i.e. Table IV(a)), we see that *normal*, *pizzicato*, *sordino*, *spiccato* and *tremolo* are generally recognizable, yet the other four techniques are more difficult to be modeled. The major ambiguities come from the pairs *flageolet* vs. *non-vibrato* (23.3%), *non-vibrato* vs. *normal* (12.7%), and *sul tasto* vs. *normal* (26.2%), among others. We observe similar confusion from the result of SG + SC (i.e. Table IV(b)). These findings imply that it is more challenging to tell the differences between timbres generated with various excitations (i.e. vibrating, non-vibrating, harmonic enhancement) and bowing positions (i.e. near the bridge, near the fingerboard), comparing to the differences resulting from finger picking, bow bouncing and muting.

[8] In contrast, the pairwise fusion SG, PH + SC leads to moderately worse F-score 0.819. As GDF and IFD can be considered as advanced version of PH, we do not report the result of fusions related to PH in Table III.

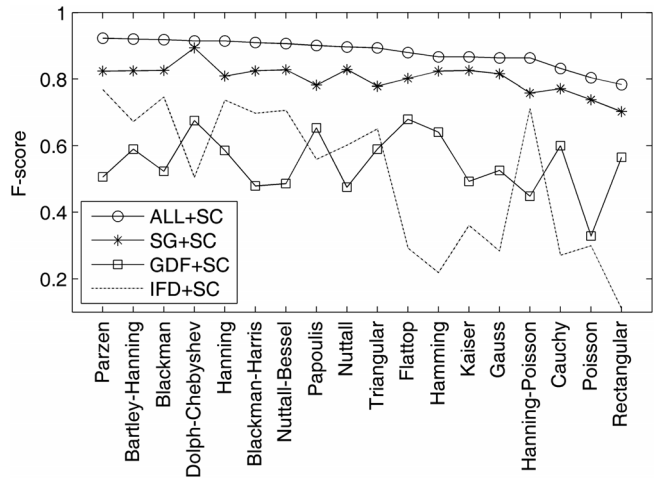


Fig. 6. Performance comparison (in F-score) of violin playing technique classification for different features and window functions.

In contrast, from the result of ALL+SC (i.e. Table IV(c)), we clearly see that adding phase derivatives resolves the aforementioned confusion without compromising the discriminability of other classes. Comparing to Table IV(b), we see improvement in precision from 55.7% to 77.0% for *flageolet*, from 50.1% to 74.4% for *non-vibrato*, and from 79.7% to 95.9% for *sul ponticello*. Subtle nuances of sound colors, such as the "intenseness" of *sul ponticello* and the "paleness" of *non-vibrato*, can be accurately modeled by sparse coding of magnitude and phase-derived spectra. Comparing Tables 4(a) and Tables (c) shows that ALL + SC outperforms TIMBRE + PCA for most playing techniques.

To compare the efficiency of the features, we also record the runtime for feature extraction for all the 2,017 violin audio samples (totaling 6,101 seconds). The runtime of extracting SG + SC is about 1.45x real time, whereas that of TIMBRE + PCA is 0.50x real time. Therefore, it is moderately more time-consuming to obtain SC features. While it is possible to accelerate SC by algorithms such as LASSO screening [44] or predictive sparse decomposition [46], this study is only concerned with the accuracy of timbre modeling.

B. Parametric Sensitivity Test

We also evaluate the effect of window functions, window length and hop size on the performance of time-frequency features for playing technique classification of violin single notes. Fig. 6 shows the result of four different SC processed time-frequency features using 18 window functions, while fixing $T = 2,048$ and $H = 10\%$. We see that GDF and IFD are more sensitive to the choice of window function than SG (see [20]), but using the Hanning window generally performs well. Although we cam achieve slightly better F-score 0.923 by using the Parzen window, we opt for using the Hanning window as it is more widely used in MIR [1].

Table V shows the result when we vary the window size T from 1,024 to 4,096 and the hop factor H from 10% to 50%,

TABLE V
PERFORMANCE COMPARISON (IN F-SCORE) OF VIOLIN PLAYING TECHNIQUE CLASSIFICATION FOR DIFFERENT WINDOW SIZES AND HOP FACTORS

hop factor	SG+SC window size			GDF+SC window size			IFD+SC window size			ALL+SC window size		
	1,024	2,048	4,096	1,024	2,048	4,096	1,024	2,048	4,096	1,024	2,048	4,096
10%	0.781	0.809	0.862	**0.720**	0.587	0.336	**0.739**	0.738	0.604	0.913	0.915	**0.927**
20%	0.782	0.857	**0.886**	0.645	0.348	0.349	0.665	0.645	0.450	0.893	0.906	0.918
50%	0.818	0.823	0.871	0.362	0.320	0.279	0.570	0.511	0.324	0.854	0.878	0.870

TABLE VI
CONFUSION MATRIX (IN %, NORMALIZED BY ROW) AND CLASS-WISE F-SCORES (IN THE RIGHTMOST COLUMN) FOR JOINT INSTRUMENT-TECHNIQUE CLASSIFICATION OF SINGLE NOTES FROM RWC, USING ALL + SC

		violin				viola				cello				contrabass				F-score
		norm	novi	pizz	supc	norm	novi	pizz	supc	norm	novi	pizz	supc	norm	novi	pizz	supc	
violin	norm	**92.3**	4.76	0.50	2.36	0.07	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.02	0.00	0.00	0.00	0.868
	novi	49.1	**48.1**	1.15	1.62	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.08	0.00	0.575
	pizz	0.00	0.00	**100**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.994
	supc	17.1	1.69	0.00	**80.7**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.848
viola	norm	0.00	0.00	0.00	0.00	**93.4**	2.92	1.03	2.64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.872
	novi	0.00	0.00	0.00	0.00	39.3	**50.2**	4.67	5.83	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.585
	pizz	0.00	0.00	0.00	0.00	0.00	0.21	**99.8**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.982
	supc	0.00	0.00	0.00	0.00	22.5	9.75	1.83	**65.9**	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.726
cello	norm	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**91.1**	5.08	2.65	1.15	0.00	0.00	0.00	0.00	0.872
	novi	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	42.9	**45.8**	3.83	7.42	0.00	0.00	0.00	0.00	0.552
	pizz	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.79	0.00	**99.2**	0.00	0.00	0.00	0.00	0.00	0.974
	supc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	14.7	0.92	1.67	**82.7**	0.00	0.00	0.00	0.00	0.852
contrabass	norm	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**87.3**	4.11	6.29	2.26	0.860
	novi	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	63.7	**19.7**	6.33	10.3	0.240
	pizz	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	2.36	0.00	**97.6**	0.05	0.921
	supc	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	27.0	11.8	7.17	**54.0**	0.598

while using the Hanning window. The best result for each feature is highlighted. We make the following observations: 1) SG prefers longer windows, yet GDF and IFD prefer shorter ones; 2) While the performance of SG is insensitive to the hop factor, GDF and IFD prefer short ones, possibly because of the aliasing effect when the frame rate (i.e. number of frames per second) is smaller than twice of the bandwidth of the window function [49];[9] 3) Although SG, GDF and IFD prefer different parameters, fusing them together (i.e.ALL+SC) yields the best F-score 0.927 by using $T = 4,096$ and $H = 10\%$, the parameter setting that comes with the finest frequency and time resolution. Despite of this, we opt for the default setting $T = 2,048$ and $H = 10\%$ as this is computationally lighter.

C. Joint Instrument-Technique Classification of Single Notes

Next, we evaluate the 16-class joint instrument-technique classification task described in Section III, using again ten-fold CV. Table VI shows the confusion matrix together with the F-scores of each class, using ALL + SC as the feature representation and linear SVM as the classifier. It can be seen that ALL + SC remains fairly effective for this task, achieving 0.770 in average F-score. Therefore, this approach captures both the instrument-level and expression-level timbre information. Moreover, we note that almost all of the misclassified labels are within one instrument and instrumental-level errors are few. For example, it is more difficult to distinguish between *non-vibrato* and *normal* sounds played by the same instrument comparing to distinguishing between violin and viola sounds of the same playing technique. This suggests that, at least for single notes, classifying playing technique is much more challenging than classifying instruments.

[9]For instance, when the sampling rate is 22,050 Hz, a Hanning window of length 2,048 has a 3 dB bandwidth of 15.5 Hz. To avoid aliasing, the frame rate needs to be higher than 31 Hz, which amounts to a period of 710 samples. Therefore, a hop factor of 50% (i.e. 1,024 samples) results in aliasing.

TABLE VII
CONFUSION MATRIX (COUNTS) AND CLASS-WISE F-SCORES FOR PLAYING TECHNIQUE CLASSIFICATION OF REAL-WORLD VIOLIN SOLOS USING DIFFERENT AUDIO FEATURES (a) TIMBRE + PCA, (b) ALL + SC

(a)

		predicted class norm	pizz	spic	supc	suta	F-score
actual class	normal	7	0	0	0	0	0.483
	pizzicato	0	8	0	0	0	0.941
	spiccato	5	1	1	0	0	0.250
	sul ponticello	6	0	0	1	0	0.250
	sul tasto	4	0	0	0	0	0.000

(b)

		predicted class norm	pizz	spic	supc	suta	F-score
actual class	normal	5	0	0	2	0	0.833
	pizzicato	0	8	0	0	0	0.941
	spiccato	0	1	0	6	0	0.000
	sul ponticello	0	0	0	7	0	0.560
	sul tasto	0	0	0	3	1	0.400

D. Real-world Violin Solo Playing Technique Classification

Finally, we evaluate playing technique classification using the real-world music collection. Instead of doing a CV, we directly use the violin single notes of the five related playing techniques from RWC to train the classifier and use the real-world dataset for testing. Tables VII(a) and Tables (b) show the resulting confusion matrices of TIMBRE + PCA and ALL + SC, respectively. As we have expected, the performance degrades much

(comparing to the previous two classification settings) due to the difference in recording environments, instrument brands, styles of the performer, among others. For $\mathrm{TIMBRE} + \mathrm{PCA}$, many samples are misclassified as *normal*, and the average F-score is 0.385, while the F-score expected by random guessing is 0.200. In contrast, $\mathrm{ALL} + \mathrm{SC}$ performs better for most classes, except for *spiccato*, and the average F-score reaches 0.547. We conjecture that RWC cannot represent all possible variations of each playing technique in real-world recordings,[10] and that a more complete training dataset would be needed for better accuracy for real-life recordings.

VI. Discussion

We discuss the main findings of this study in this section. Section V-A compares different audio features and processing methods for playing technique classification of violin single notes. For audio features extracted by the MIRtoolbox (i.e. TIMBRE), we obtain the best average F-score of 0.854 by using PCA for feature processing, $\mathrm{mean} + \mathrm{std}$ pooling for temporal aggregation, z-score normalization as a post-processing, and RBF kernel SVM as the classifier. Neither VQ nor SC is useful for TIMBRE. In contrast, for time-frequency features, SC plays an important role in promoting the discriminability of these features. Fusing SC-processed log magnitude spectrum, GDF and IFD features (i.e. $\mathrm{ALL}+\mathrm{SC}$ leads to average F-score 0.915, which is significantly better than the result of $\mathrm{TIMBRE}+\mathrm{PCA}$. The per-class precision is also improved. For $\mathrm{ALL} + \mathrm{SC}$, we use ODL to learn a 1,024-codeword dictionary, mean pooling for temporal aggregation, square root power normalization as a post-processing, and linear kernel SVM as the classifier. As reported in Section 5.2, we recommend using 2,048-point Hanning window with hop factor 20%, though slightly better accuracy can be obtained by using other parameter settings at increased computational cost. Appendix 7 provides more discussion on the effect of playing techniques on the phase spectra and phase derivatives.

We find that it is relatively easier to model *normal*, *pizzicato*, *sordino*, *spiccato* and *tremolo* for the violin single notes, and that adding phase-related features greatly improves the modeling of the remaining, more challenging, cases. Importantly, our study validates the effectiveness of the sparse modeling of magnitude and phase-derived spectra in characterizing subtle musical timbre, even though it is based on primitive time-frequency features and is computed without explicitly exploiting music knowledge for feature design.

Sections V-C and V-D show that classifying playing techniques is more challenging than classifying instruments, and that the single-note violin samples we have collected from the RWC database might not be able to represent all possible variations of the playing techniques in real-world violin solos. While future works are needed to improve the accuracy, our evaluation shows that $\mathrm{ALL} + \mathrm{SC}$ captures both the instrument-level and expression-level timbre information and leads to promising result (i.e. F-score 0.547) in the real-world setting.

[10]For example, a player can perform *spiccato* with a variety of bouncing forces, speeds and tilting angles, and perform *sul ponticello* and *sul tasto* with a variety of bowing position.

VII. Conclusion

In this paper, we have presented a systematic evaluation of various audio features and processing methods for three different settings of multi-class playing technique classification, using audio recordings of bowed string instruments. It extends the well-studied musical instrument classification problem to the level of individual playing techniques, which is relatively unexplored. Moreover, the study illustrates the importance of phase derivatives in discriminating subtle differences in musical timbre, and shows that sparse coding is an effective means to mining useful patterns from such primitive time-frequency representations.

Although the present study might be at best preliminary, we hope it can call for more attention towards the expression level of musical timbre and the investigation of phase related features for music classification.

Appendix A
Phase, GDF and IFD

A non-stationary, quasi-periodic signal can be generally expressed as a series of band-limited signals:

$$x(t,\omega) = \sum_{k=1}^{K} A_k(t,\omega)\, e^{j(\omega_k t + \phi_k(t,\omega))}, \tag{11}$$

where $A_k(t)$ denotes the amplitude and $\phi_k(t)$ is the phase of the k-th harmonics ω_k. All of these parameters, including the phase term $\phi_k(t)$, contribute to timbre information in either instrument-level or expression level. However, because the oscillating term $e^{j\omega_k t}$ dominates the relatively slow-varying terms $A_k(t)$ and $\phi_k(t)$, directly using the phase spectra (i.e. $\omega_k t + \phi_k(t,\omega)$) as the feature may not work well. Taking derivatives of phase with respect to either time or frequency help eliminate the oscillating term and reveal useful information. For example, as described in Section II, the slope of the saw-tooth patterns of the phase variation over time displayed in the third row of Fig. 3 carries timbre information.

Spectrogram characterizes the energy distribution and, more importantly, the harmonic pattern of a signal. In contrast, IFD and GDF characterize the local behaviors of phase with respect to time and frequency, respectively. In discrete implementation, IFD in Eq. (5) indicates the deviation the instantaneous frequency from the nearest bin frequency (the time derivative of the phase term $\phi_k(t)$ is only the frequency term which is not included in the $e^{j\omega_k t}$ term). Therefore, we get a good calibration by using IFD, especially under low spectral resolution or high spectral leakage due to windowing.

For a bowed string instrument, the response curves of SG and GDF accurately reflect the behaviors of the *resonance* (transmission poles) and *antiresonance* (transmission zeros) modes [18], as determined by the physical structure of the instrument (e.g. string length and the cavity shape) and the extent to which the input energy excites the modes. While it is not easy to analyze the entire physics of an instrument, we use a simple resonator model to illustrate how GDF is related to the resonant modes of the signal. Fig. 7 compares two resonance peaks: (a) a resonance accompanied with an antiresonance and (b) a simple

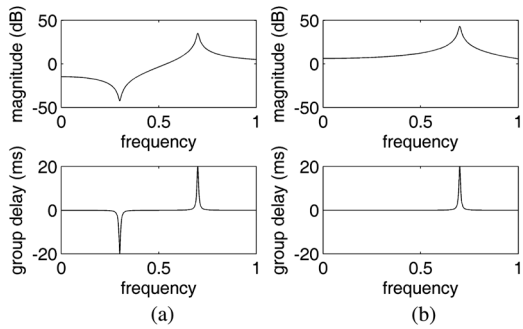


Fig. 7. The (from top to bottom) magnitude spectra and group delay functions of (a) a signal with a resonance and an antiresonance (and non-minimal-phase response) and (b) a signal with a simple resonance.

resonance. As the first row shows, it might not be easy to identify the presence of an antiresonance from the SG when there are random noises. In contrast, due to the prominent negative peak in the GDF shown in the second row, it is easier to discriminate between the two signals (a) and (b) using GDF. More discussions on the resonance of instruments can be found in [50].

The effect of playing technique variation on phase has also been studied in acoustic research. For example, Beauchamp [51] described the temporal variation behaviors of amplitudes and phases with respect to the formants of violin tones, demonstrating significant difference between *vibrato* and *non-vibrato* by harmonic phase analysis. Mellody and Wakefield [52] investigated the amplitude and frequency modulation characteristics of violin vibrato sounds based on high-resolution time-frequency analysis and compared the perceptual importance of frequency and amplitude modulation by subjective tests. The phase spectra also carry information about the modulation effects such as *jitter* (i.e. the periodical variation of frequency) and *shimmer* (i.e. the periodical variation of amplitude) [53].

Phase has also been considered important in speech processing for several years [20]–[23] and many advanced phase-related features have been proposed. For example, Murthy *et al.* proposed the modified group delay function, capable of eliminating the transmission zeros located near the unit circle in the z-plane while preserving the original GDF behaviors [21]. For simplicity we adopt the fundamental approaches to computed GDF and IFD in this work.

APPENDIX B
ONLINE DICTIONARY LEARNING

Online dictionary learning (ODL) is a first-order stochastic gradient descent algorithm proposed by Mairal *et al.*[33] to solve the following optimization problem,

$$\begin{aligned} \arg\min_{\mathbf{D},\mathbf{A}} \quad & \frac{1}{n}\sum_{i=1}^{n}\left(\frac{1}{2}\|\mathbf{x}_i-\mathbf{D}\boldsymbol{\alpha}_i\|_2^2+\lambda\|\boldsymbol{\alpha}_i\|_1\right), \\ \text{subject to} \quad & \mathbf{d}_j^T\mathbf{d}_j \le 1, \forall j=1,\ldots,k, \end{aligned} \tag{12}$$

where the parameter λ can be set to $m^{-1/2}$ as in [33], α_i is the encoding result of the i-th frame-level feature vector $\mathbf{x}_i$, which can be log-scaled spectrum, GDF or IFD, and $\mathbf{A}=[\alpha_1,\ldots,\alpha_n]\in\mathbb{R}^{k\times n}$. A natural solution to this problem is to solve the two variables $\mathbf{D}$ and $\mathbf{A}$ by minimizing one while maintaining the other fixed. The optimization of $\mathbf{D}$ uses block coordinate descent with warm restarts, which aggregates the previous information calculated during the previous steps of the algorithm. Optimizing $\mathbf{A}$ involves a typical LASSO problem (i.e. Eq. (9)). Thanks to its low memory consumption and computational cost, ODL is more efficient than standard second-order matrix factorization algorithms such as K-SVD [32] for large-scale problems. Using a dictionary learned from ODL for SC has also been shown effective [43].

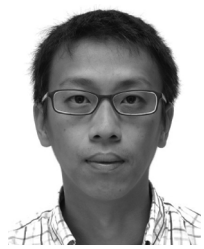
ACKNOWLEDGMENT

The authors thank the anonymous reviewers for their valuable suggestions that help improve the quality of this paper.

REFERENCES

[1] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1088–1110, Dec. 2011.
[2] S. Essid, G. Richard, and B. David, "Instrument recognition in polyphonic music based on automatic taxonomies," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 1, pp. 68–80, Jan. 2006.
[3] N. Chetry and M. Sandler, "Linear predictive models for musical instrument identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2006, pp. 225–228.
[4] J. J. Burred, A. Röbel, and T. Sikora, "Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 3, pp. 663–674, Mar. 2010.
[5] J. G. A. Barbedo and G. Tzanetakis, "Musical instrument classification using individual partials," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 1, pp. 111–122, Jan. 2011.
[6] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Canadas-Quesada, "Musical instrument sound multi-excitation model for non-negative spectrogram factorization," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1144–1158, Dec. 2011.
[7] S. Scholler and H. Purwins, "Sparse approximations for drum sound classification," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 5, pp. 933–940, Sep. 2011.
[8] P. Leveau, D. Sodoyer, and L. Daudet, "Automatic instrument recognition in a polyphonic mixture using sparse representations," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2007, pp. 233–236.
[9] F. Fuhrmann, "Automatic musical instrument recognition from polyphonic music audio signals," Ph.D. dissertation, Univ. Pompeu Fabra, Barcelona, Spain, 2012.
[10] A. Zlatintsi and P. Maragos, "Multiscale fractal analysis of musical instrument signals with application to recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 21, no. 4, pp. 737–748, Apr. 2013.
[11] L.-F. Yu, L. Su, and Y. H. Yang, "Sparse cepstral codes and power scale for instrument identification," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 7460–7464.
[12] D. D. Boyden, *The History of Violin Playing from Its Origins to 1761 and Its Relationship to the Violin and Violin Music*. Oxford, U.K.: Oxford Univ. Press, 1965.
[13] R. Stowell, *Violin Technique and Performance Practice in the Late Eighteenth and Early Nineteenth Centuries (Cambridge Musical Texts and Monographs)*. Cambridge, U.K.: Cambridge Univ. Press, 1990.
[14] S. Adler, *The Study of Orchestration*. New York, NY, USA: W. W. Norton and Co. Inc., 2002.
[15] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka, "RWC music database: Music genre database and musical instrument sound database," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2003, pp. 229–230.
[16] F. Auger and P. Flandrin, "Improving the readability of time-frequency and time-scale representations by the method of reassigment," *IEEE Trans. Signal Process.*, vol. 43, no. 5, pp. 1068–1089, May 1995.
[17] K. R. Fitz and S. A. Fulop, "A unified theory of time-frequency reassignment," *CoRR*, vol. abs/0903.3080, 2009.
[18] J. Bechhoefer, "Kramers-Kronig, bode, and the meaning of zero," *Amer. J. Phys.*, vol. 79, no. 10, pp. 1053–1060, 2011.
[19] D. Ellis, "A phase vocoder in Matlab," , 2002 [Online]. Available: http://www.ee.columbia.edu/dpwe/resources/matlab/pvoc/
[20] K. K. Paliwal and L. D. Alsteris, "Further intelligibility results from human listening tests using the short-time phase spectrum," *Speech Commun.*, vol. 48, pp. 727–736, 2006.

[21] R. M. Hegde, H. A. Murthy, and V. R. R. Gadde, "Significance of the modified group delay feature in speech recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 15, no. 1, pp. 190–202, Jan 2007.
[22] I. McCowan, D. Dean, M. McLaren, R. Vogt, and S. Sridharan, "The delta-phase spectrum with application to voice activity detection and speaker recognition," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 19, no. 7, pp. 2026–2038, Sep. 2011.
[23] E. Loweimi, S. M. Ahadi, and T. Drugman, "A new phase-based feature representation for robust speech recognition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 7155–7159.
[24] M. Goto, "A robust predominant-F0 estimation method for real-time detection of melody and bass lines in cd recordings," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2000, pp. 757–760.
[25] A. Holzapfel, Y. Stylianou, A. Gedik, and B. Bozkurt, "Three dimensions of pitched instrument onset detection," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 6, pp. 1517–1527, Aug. 2010.
[26] L. Su and Y.-H. Yang, "Power-scaled spectral flux and peak-valley group-delay methods for robust musical onset detection," in *Proc. Sound Music Comput. Conf.*, 2014.
[27] O. Lartillot and P. Toiviainen, "A Matlab toolbox for musical feature extraction from audio," in *Proc. Int. Conf. Digital Audio Effects*, 2007, pp. 237–244.
[28] L. Su and Y. H. Yang, "Sparse modeling for artist identification: Exploiting phase information and vocal separation," in *Proc. Int. Soc. Music Inf. Retrieval Conf.*, 2013, pp. 349–354.
[29] B. Yegnanarayana and H. A. Murthy, "Significance of group delay functions in spectrum estimation," *IEEE Trans. Signal Process.*, vol. 40, no. 9, pp. 2281–2289, Sep. 1992.
[30] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *J. R. Statist. Soc.*, vol. 58, pp. 267–288, 1996.
[31] B. Efron, T. Hastie, L. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Statist.*, vol. 32, pp. 407–499, 2004.
[32] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.
[33] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 689–696.
[34] C.-C. Chang and C.-J. Lin, *LIBSVM: A Library for Support Vector Machines,* 2001*,* in .
[35] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.
[36] Z. Fu, G. Lu, K. Ting, and D. Zhang, "A survey of audio-based music classification and annotation," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 303–319, Apr. 2011.
[37] R. Loughran, J. Walker, M. O'Neill, and M. O'Farrell, "Musical instrument identification using principal component analysis and multilayered perceptrons," in *Proc. Int. Conf. IEEE Audio, Lang. Image Process. (ICALIP '08)*, 2008, pp. 643–648.
[38] F. Auger, P. Flandrin, P. Goncalves, and O. Lemoine*, Time-Frequency Toolbox for Use with MATLAB*, 1996 [Online]. Available: http://tftb.nongnu.org
[39] M. R. Šikonja and I. Kononenko, "Theoretical and empirical analysis of ReliefF and RReliefF," *Mach. Learn.*, vol. 53, pp. 23–69, 2003.
[40] [Online]. Available: http://www.mathworks.com/products/statistics/
[41] A. Gersho and R. M. Gray*, Vector quantization and signal compression*. Norwell, MA, USA: Kluwer, 1991.
[42] E. C. Smith and M. S. Lewicki, "Efficient auditory coding," *Nature*, vol. 439, no. 7079, pp. 978–982, 2006.
[43] L. Su, C.-C. M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang, "A systematic evaluation of the bag-of-frames representation for music information retrieval," *IEEE Trans. Multimedia*, vol. 16, no. 5, pp. 1188–1200, Aug. 2014.
[44] P.-K. Jao, C.-C. M. Yeh, and Y.-H. Yang, "Modified LASSO screening for audio word-based music classification using large-scale dictionary," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2014, pp. 5207–5211.
[45] Y. H. Yang, "Towards real-time music auto-tagging using sparse features," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2013, pp. 1–6.
[46] A. Coates and A. Ng, "The importance of encoding versus training with sparse coding and vector quantization," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 921–928.
[47] I. Tosic and P. Frossard, "Dictionary learning: What is the right representation for my signal?," *IEEE Signal Process. Mag.*, vol. 28, no. 2, pp. 27–38, Mar. 2011.
[48] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.
[49] J. B. Allen and L. R. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proc. IEEE*, vol. 65, no. 11, pp. 1558–1564, Nov. 1977.
[50] N. N. H. Fletcher and T. D. Rossing*, The Physics of Musical Instruments*. New York, NY, USA: Springer, 1998.
[51] J. W. Beauchamp, "Time-variant spectra of violin tones," *J. Acoust. Soc. Amer.*, vol. 56, no. 3, pp. 995–1004, 1973.
[52] M. Mellody and G. H. Wakefield, "The time-frequency characteristics of violin vibrato: Modal distribution analysis and synthesis," *J. Acoust. Soc. Amer.*, vol. 107, no. 1, pp. 598–611, 2000.
[53] M. Farrús, J. Hernando, and P. Ejarque, "Jitter and shimmer measurements for speaker recognition," in *Proc. Interspeech*, 2007, pp. 778–781.



Li Su (S'08–M'13) received the Ph.D. degree in communication engineering from National Taiwan University, Taiwan, in 2012. Since May 2012, he has been a Postdoctoral Research Fellow in Research Center for Information Technology Innovation, Academia Sinica, Taiwan. His research interests include music information retrieval, machine learning, signal processing and filter design. He is the Tutorial Chair of the International Society for Music Information Retrieval Conference (ISMIR 2014).

Hsin-Ming Lin received the master degree in music technology from National Chiao Tung University, Taiwan, in 2011. He is currently a Ph.D. student in the Computer Music program, Department of Music, University of California, San Diego. His research interests include algorithmic composition, music information retrieval, and interactive music.

Yi-Hsuan Yang (M'11) received the Ph.D. degree in communication engineering from National Taiwan University in 2010. Since 2011, he has been affiliated with Academia Sinica as an Assistant Research Fellow. His research interests include music information retrieval, machine learning and affective computing. In 2014, he serve as a Technical Program Co-Chair of ISMIR and a Guest Editor of the IEEE TRANSACTIONS ON AFFECTIVE COMPUTING.